

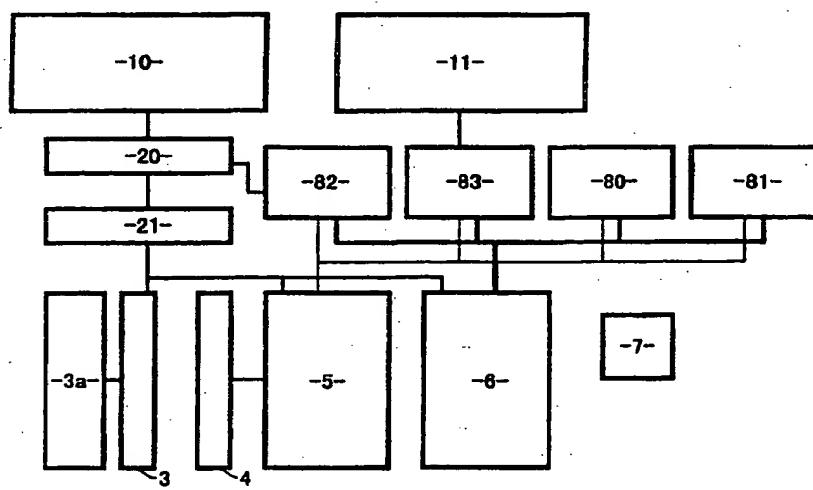
PCT

特許協力条約に基づいて公開された国際出願

(51) 国際特許分類6 G06F 9/38, 15/82	A1	(11) 国際公開番号 WO99/27439
		(43) 国際公開日 1999年6月3日(03.06.99)
(21) 国際出願番号 PCT/JP98/05230		
(22) 国際出願日 1998年11月19日(19.11.98)		
(30) 優先権データ 特願平9/362473 特願平10/282118	1997年11月20日(20.11.97) 1998年10月5日(05.10.98)	JP JP
(71) 出願人；および 関 一(SEKI, Hajime)[JP/JP] 〒790-0848 愛媛県松山市道後喜多町4番38号 Ehime, (JP)		(81) 指定国 AU, BG, BR, CA, CN, CU, CZ, HU, ID, IL, IS, JP, KR, MX, NO, NZ, PL, SG, SK, TR, UA, US, VN, ARIPO特許(GH, GM, KE, LS, MW, SD, SZ, UG, ZW), ヨーラシア特許(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), 欧州特許(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI特許(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
		添付公開書類 国際調査報告書

(54) Title: COMPUTER SYSTEM

(54) 発明の名称 計算機システム



(57) Abstract

A computer system for processing a program described in the machine language of a stack machine, comprising a data cache (11), an integrated register file (6) in which data is written in each of entries, an advanced pointer stack (3) and a completion pointer stack (4) in which the addresses of the entries of the integrated register file are written into entries, an instruction queue (5) having the construction of an FIFO in which the contents of each of instructions are written into each entry, arithmetic units (80 and 81), and a loading/storing unit (83). When the instruction held in the first entry of the instruction queue can be completed or has been completed, the completion pointer stack is operated so as to execute the same operation of the advanced pointer stack when the held instruction is decoded on the basis of the contents of the first entry of the instruction queue, and the first entry is eliminated from the instruction queue.

データ・キャッシュ(11)と各々のエントリにデータが書き込まれるようになっている統合レジスタ・ファイル(6)と、各々のエントリに統合レジスタ・ファイルのエントリのアドレスが書き込まれるようになっている前進ポインタ・スタック(3)及び完了ポインタ・スタック(4)と、各々のエントリに個々の命令の内容が書き込まれるようになっているFIFOの構成となっている命令キュー(5)と、演算ユニット(80), (81)と、ロード/ストア・ユニット(83)を具備する、スタックマシンの機械語で記述されたプログラムを処理する計算機システムである。

命令キューの先頭のエントリにおいて保持されている命令の完了が可能である、あるいはそうなると、その命令キューの先頭のエントリの内容に基づき、保持されている命令がデコードされた際の前進ポインタ・スタックの動作を再現すべく完了ポインタ・スタックを操作し、命令キューからその先頭のエントリを除外するようになっている。

PCTに基づいて公開される国際出願のパンフレット第一頁に掲載されたPCT加盟国を同定するために使用されるコード(参考情報)

AE アラブ首長国連邦	ES スペイン	LI リヒテンシュタイン	SG シンガポール
AL アルベニア	FI フィンランド	LK スリ・ランカ	SI スロヴェニア
AM アルメニア	FR フランス	LR リベリア	SK スロヴァキア
AT オーストリア	GA ガボン	LS レソト	SL シエラ・レオネ
AU オーストラリア	GB 英国	LT リトアニア	SN ゼネガル
AZ アゼルバイジャン	GD グレナダ	LU ルクセンブルグ	SZ スワジランド
BA ボスニア・ヘルツェゴビナ	GE グルジア	LV ラトヴィア	TD チャード
BB バルバドス	GH ガーナ	MC モナコ	TG トーゴー
BE ベルギー	GM ガンビア	MD モルドバ	TJ タジキスタン
BF ブルガリア・ファン	GN ギニア	MG マダガスカル	TM トルクメニスタン
BG ブルガリア	GW ギニア・ビサオ	MK マケドニア旧ユーゴスラヴィア	TR トルコ
BJ ベナン	GR ギリシャ	共和国	TT トリニダード・トバゴ
BR ブラジル	HR クロアチア	ML モンゴル	UA ウクライナ
BY ベラルーシ	HU ハンガリー	MR モーリタニア	UG ウガンダ
CA カナダ	ID インドネシア	MW マラウイ	US 米国
CF 中央アフリカ	IE アイルランド	MX メキシコ	UZ ウズベキスタン
CG コンゴー	IL イスラエル	NE ニジエール	VN ヴィエトナム
CH スイス	IN インド	NL オランダ	YU ユーゴースラビア
CI コートジボアール	IS アイスランド	NO ノルウェー	ZA 南アフリカ共和国
CM カメルーン	JP 日本	NZ ニュー・ジーランド	ZW ジンバブエ
CN 中国	KE ケニア	PL ポーランド	
CU キューバ	KG キルギスタン	PT ポルトガル	
CY キプロス	KP 北朝鮮	RO ルーマニア	
CZ チェコ	KR 韓国	RU ロシア	
DE ドイツ	KZ カザフスタン	SD スーダン	
DK デンマーク	LC セントルシア	SE スウェーデン	
EE エストニア			

明細書

計算機システム

技術分野

本発明は、スタックマシンの機械語で記述されたプログラムを高速で
5 处理する新規な構成の計算機システムに関するものである。

背景技術

従来、スタックマシンにおいては、命令の実行は、基本的にプログラム
10 上の順序通り（in-order）に行われるものであった。すなわち、スタ
ックマシンにおける演算命令は、オペランド・スタックからソース・デー
タをポップし、演算を実行し、その演算結果をオペランド・スタックに
15 プッシュするというような動作を指示するものであるが、このような命
令の連鎖として書かれたプログラムを逐次的に実行するのである。

このような従来のスタックマシンにおいては、命令をプログラム上の
順序通り（in-order）に実行するので、制御構造が単純なもので済むと
15 いう利点があるが、処理速度が制約を受けるという問題点があった。

そこで、スタックマシンの機械語で記述されたプログラムを
out-of-orderで処理するような計算機方式が考案された。例えば、日本特
20 公平2-260082号、米国特許第5522051号や、米国特許第
5333320号及び米国特許第5765014号におけるプロセッサ
要素がある。これらの明細書に示されるプロセッサは、処理性能の向上
という点で十分ではない上に、正確な例外処理を保証する上で問題があ
った。

本発明は、上記問題点を解決するため創案されたものであり、正確な

例外処理を保証しつつ、スタックマシンの機械語で記述されたプログラムを *out-of-order* でより効率的に処理する計算機システムを提供することを目的としている。

5 発明の開示

本発明による計算機システムは、データ・キャッシュと、各々のエントリにデータが書き込まれるようになっている統合レジスタ・ファイルと、各々のエントリに統合レジスタ・ファイルのエントリのアドレスが書き込まれるようになっているスタックの構成となっている前進ポインタ・スタック及び完了ポインタ・スタックと、各々のエントリに個々の命令の内容が書き込まれるようになっているFIFOキューの構成となっている命令キューと、演算を実行するようになっている演算ユニットと、データ・キャッシュ及び統合レジスタ・ファイルにアクセスできるようになっているロード/ストア・ユニットとを具備する。

従来のスタックマシンにおいて、スタックが , word1, word2, word3, word4 (右端がスタックトップ) となっている状態は、本発明による計算機システムにおいて、ポインタ・スタックが , ⟨a⟩, ⟨b⟩, ⟨c⟩, ⟨d⟩ (右端がスタックトップ) で、エントリ・アドレスが ⟨a⟩, ⟨b⟩, ⟨c⟩ 及び ⟨d⟩ である統合レジスタ・ファイルの各エントリに、それぞれ word1, word2, word3 及び word4 が保持されている状態に対応する。

本発明の計算機システムにおいては、命令がデコードされるごとに、その命令の内容に応じて前進ポインタ・スタック及び統合レジスタ・ファイルを操作すると共にその命令の内容を命令キューに書き込むようになっている。この際、命令に含まれるオペランド・スタックに対するスタック操作が、前進ポインタ・スタックに対して同様に適用される。ここ

で、1語のデータのオペランド・スタックへのプッシュ操作を、本発明の計算機システムにおいてエミュレートするには、割り付けられていない統合レジスタ・ファイルの1エントリをそのデータを保持すべく割り付け、そのエントリのアドレスを前進ポインタ・スタックにプッシュすればよい。

即ち、オペランド・スタックに対するポップ操作を含む命令がデコードされた場合には、ポップすべき語数と同じ数だけ統合レジスタ・ファイルのエントリのアドレスを前進ポインタ・スタックからポップする。オペランド・スタックに対するプッシュ操作を含む命令がデコードされた場合には、プッシュすべき語数と同じ数だけ割り付けられていない統合レジスタ・ファイルのエントリを割り付け、上記割り付けた統合レジスタ・ファイルのエントリのアドレスを前進ポインタ・スタックにプッシュする。さらに、デコードされた命令の内容を、ポップ/プッシュ操作を伴う命令の場合にはポップ/プッシュされる統合レジスタ・ファイルのエントリのアドレスと共に、命令キューに書き込むようになっている。

命令キューに保持されている未実行の命令は、データ駆動（各々の動作が、必要なすべてのソース・データが揃い実行可能となった時点で実行されること）の原理に基づき *out-of-order* で処理されるようになっている。

例えば、命令キューにおいて、演算命令を書き込みの内容とし、必要なソース・データが全て統合レジスタ・ファイルに書き込み済みとなっているエントリがあり、演算ユニットが利用できる状態であれば、その演算の実行を開始する。演算の実行が正常に終了すれば、演算結果をデスティネーションである統合レジスタ・ファイルのエントリに書き込む。

命令キューの先頭のエントリにおいて保持されている命令の完了が可能である、あるいはそうなると、その命令キューの先頭のエントリの内

容に基づき、保持されている命令がデコードされた際の前進ポインタ・
スタックの動作を再現すべく完了ポインタ・スタックを操作し、命令キ
ューからその先頭のエントリを除外し、ポップ操作によって完了ポイン
タ・スタックにおけるアドレスの保持が無くなった統合レジスタ・ファイ
ルのエントリの割り付けを解除するようになっている。

図面の簡単な説明

第1図は、本発明にかかる好ましい計算機システムの基本構成を示す
ブロック図、第2図は、前進ポインタ・スタック及び完了ポインタ・スタ
ックの構成を示す図、第3図は、後述する本発明第1実施例における統
合レジスタ・ファイルの各々のエントリの詳細な構成を示す図、第4図
は、命令キューの構成を示す図、第5図は、命令キューの各々のエント
リの詳細な構成を示す図、第6図～第12図は、本発明第1実施例にお
ける一動作例の、サイクル毎の前進ポインタ・スタック、完了ポインタ・
スタック、命令キュー及び統合レジスタ・ファイルの内容を具体的に示
した説明図、第13図は、本発明第1実施例において、1サイクル当たり
3命令までデコードできるような構成をとる場合に、プログラムがどの
ように変換されるかを具体的に示す図表、第14図は、後述する本発明
第2実施例における統合レジスタ・ファイルの各々のエントリの詳細な
構成を示す図、第15図～第21図は、本発明第2実施例における一動
作例の、サイクル毎の前進ポインタ・スタック、完了ポインタ・スタック、
命令キュー及び統合レジスタ・ファイルの内容を具体的に示した説明図
である。

発明を実施するための最良の形態

以下に、本発明にかかる好ましい計算機システムについて、図面を参

照しながら説明する。なお、以下に述べる本発明による計算機システムの実施例は、Java Virtual Machine (Java VM) で規定されるスタックマシンの基本的な命令をハードウェアで実行するものである。すなわち、データ語長を 32 ビットとして、これを単位にロード／ストア及び算術論理演算等の演算を行う。従って、例えば、倍長語の間での算術演算は、2 語づつ合せて 4 語のソース・データをもとに 2 語の演算結果を生ずる。

従来のスタックマシンにおける、語の単位でデータがプッシュ／ポップされるようになっているスタックは、後述するポインタ・スタックと区別するために、以降では、ワード・スタックと呼ぶことにする。

Java VM にはもともとハードウェアで実行することを想定していない複雑な命令が含まれるが、以下に述べる本発明による計算機システムの実施例は、次のような基本的な命令をハードウェアで実行するものとする。

(a) 即値データのオペランド・スタックへのプッシュ命令

bipush, sipush, aconst_null, icanst_m1, icanst_{i}, fconst_{f}, lconst_{l},
dconst_{d}

(b) 変数データのオペランド・スタックへのロード命令

ldc1, ldc2, iload, iload_{n}, float, float_{n}, aload, aload_{n}, ldc2w, iload,
lload_{n}, dload, dload_{n}, iaload, laload, faload, daload, aaload, baload,
caload, saload

(c) オペランド・スタック上のデータの変数へのストア命令

istore, istore_{n}, fstore, fstore_{n}, astore, astore_{n}, lstore, lstore_{n},
dstore, dstore_{n}, iastore, lastore, fastore, dastore, aastore, bastore, castore,
sastore

(d) 演算命令

(d-1) 算術演算命令

iadd, ladd, fadd, dadd, isub, lsub, fsub, dsub, imul, lmul, fmul, dmul, idiv, ldiv, fdiv, ddiv, irem, lrem, frem, drem, ineg, lneg, fneg, dneg

(d - 2) 論理演算命令

ishl, ishr, iushr, lshl, lshr, lushr, iand, land, ior, lor, ixor, lxor

5 (d - 3) 変換演算命令

i2l, i2f, i2d, l2i, l2f, l2d, f2l, f2d, d2i, d2f, int2byte, int2char, int2short

(d - 4) 比較演算命令

lcmp, fcmpl, fcmpg, dcmpl, dcmpg

(e) オペランド・スタックの操作命令

10 pop, pop2, dup, dup2, dup_x1, dup2_x1, dup_x2, dup2_x2, swap

(f) 分岐命令

ifeq, ifnull, iflt, ifle, ifne, ifnonnull, ifgt, ifge, if_icmpne, if_icmplt, if_icmpgt, if_icmple, if_icmpge, goto, goto_w

以降、特にことわらない限り、「命令」とは上に挙げた命令のいずれかを意味するものとする。

15 以下に、オペランド・スタックの操作命令の処理方法の異なる、第1及び第2の2つの実施例について説明する。

まず、本発明第1実施例の計算機システムについて説明する。

第1図は計算機システムのブロック図であって、10は命令キャッシュ、11はデータ・キャッシュ、20は命令フェッチ・ユニット、21は命令デコード・設定ユニット、3は前進ポインタ・スタック、3aは前進ポインタ・スタック履歴ファイル、4は完了ポインタ・スタック、5は命令キュー、6は統合レジスタ・ファイル、7はフリー・リスト、80及び81は各々演算ユニット0及び1、82は分岐ユニット、83はロード／ストア・ユニットを表している。

次に、本発明第1実施例の計算機システムの各構成要素ごとにその詳

細な構成を説明する。

(A) 命令フェッチ・ユニット

命令フェッチ・ユニットは、図示していないプログラムカウンタ (pc レジスタ) を具備しており、命令キャッシュから命令をフェッチし、命令デコード・設定ユニットに渡す。分岐の予測や分岐の実行も担う。

(B) 命令デコード・設定ユニット

命令デコード・設定ユニットは、命令フェッチ・ユニットから渡された命令のデコードを行い、命令に含まれる演算等がデータ駆動で実行されるように、後述する前進ポインタ・スタック、命令キュー及び統合レジスタ・ファイル等を設定するための各種信号を発生する。

(C) ポインタ・スタック

ポインタ・スタックは、各々のエントリに統合レジスタ・ファイルのエントリのアドレスが書き込まれるようになっているスタックの構成となっている。

従来のスタックマシンにおいて、ワード・スタックが , word1, word2, word3, word4 (右端がスタックトップ) となっている状態は、本発明による計算機システムにおいて、ポインタ・スタックが , <a> , , <c> , <d> (右端がスタックトップ) で、エントリ・アドレスが <a> , , <c> 及び <d> である統合レジスタ・ファイルの各エントリに、それぞれ word1, word2, word3 及び word4 が保持されている状態に対応する。

本発明の計算機システムは、前進ポインタ・スタック (A P S ; Advanced Pointer Stack) と完了ポインタ・スタック (C P S ; Completed Pointer Stack) の2つのポインタ・スタックを具備する。

本発明の計算機システムにおいては、個々の命令がデコードされるごとに、前進ポインタ・スタック (以下では A P S で示す) 及び統合レジ

スタ・ファイルを操作すると共に命令の内容を命令キューに書き込むことにより、命令に含まれる演算等がデータ駆動で実行されるべく設定される。すなわち、前進ポインタ・スタックはデコード・設定済みの全ての命令によるスタック操作を反映している。

5 他方、完了ポインタ・スタック（以下ではCPSで示す）は、プログラム上の順番で完了済みの全ての命令によるスタック操作を反映するものである。本発明の計算機システムはデータ駆動の原理に基づく 10 out-of-order 実行を可能とするものであるが、完了ポインタ・スタックは、正確な例外処理を保証するため、プログラムが in-order で実行された場合の状態を構成するために存在するものである。

10 ポインタ・スタック及び統合レジスタ・ファイルのエントリ数は限られたものであるので、ワード・スタックが成長すると、ポインタ・スタック及び統合レジスタ・ファイルを用いてスタック・トップ近傍の部分しか保持できない。本実施例の計算機システムにおいては、ワード・スタックの残りの部分はデータ・キャッシュに格納されるようになっている。そのため、各ポインタ・スタックは循環型のバッファの構成となっており、 15 プッシュ・ポインタとボトム・ポインタと呼ぶ2つのレジスタが各々存在する。プッシュ・ポインタは、統合レジスタ・ファイルのエントリのアドレスを保持する最上位のエントリの1つ上を示す。ボトム・ポインタは、統合レジスタ・ファイルのエントリのアドレスを保持する最下位のエントリを示す。ボトム・ポインタの値からプッシュ・ポインタの値を引くことで、ポインタ・スタックに何エントリの空きがあるかがわかる。初期 20 状態においては、プッシュ・ポインタ及びボトム・ポインタの各々の値は共に0となっている。

25 第2図は、本実施例の計算機システムにおける各ポインタ・スタックと各プッシュ・ポインタ及びボトム・ポインタの関係を示す説明図である。

る。2つのポインタ・スタックAPS3及びCPS4は同数のエントリを有し、各ポインタ・スタックで各々のエントリに下から順に0、1、2、…とアドレスが付けられているものとする。縦線が施されているエントリは統合レジスタ・ファイルのエントリのアドレスを保持しているものとする。第2図に示すように、プッシュ・ポインタは、APS及びCPSの各々に対して設けられており、それぞれPP_OFAPS及びPP_OF_CPSと名付けている。他方、ボトム・ポインタは1つだけ設けられており、これがAPS及びCPSで共用される。これをBP_OF_PSと名付けている。

APSとCPSの間には、エントリの数だけ比較回路が設けられており、APS及びCPSの同じエントリ・アドレスにある（第2図において水平に並ぶ）エントリの間でその内容が比較されるようになっている。

命令に含まれるオペランド・スタックに対する1語分のプッシュ操作に対応して割り付けられる統合レジスタ・ファイルの1エントリのアドレスをAPSのPP_OFAPSで示されるエントリに書き込み、PP_OFAPSの値に1を加えるようになっている。逆に、命令に含まれるオペランド・スタックに対する1語分のポップ操作に対応して、PP_OFAPSの値から1を引くようになっている。CPSとPP_OF_CPSに関しても同様である。

BP_OF_PSで示されるエントリの内容がAPSとCPSで一致する場合には、その2つのポインタ・スタックで一致する内容で示される統合レジスタ・ファイルのエントリに書き込まれている1語分のデータをデータ・キャッシュにストア（Spill）することができる。その際、BP_OF_PSの値に1を加えるようになっている。逆に、データ・キャッシュにストア（Spill）したデータを統合レジスタ・ファイルにロード（Fill）する場合には、最後にストア（Spill）した1語分のデータに対し、フリー・リ

ストに登録されている統合レジスタ・ファイルの1エントリを割り付けてそのデータを書き込み、その統合レジスタ・ファイルのエントリのアドレスをAPS及びCPSのBP_OF_PSで示されるエントリの1つ下に各々書き込み、BP_OF_PSの値から1を引くようになっている。

5 本実施例の計算機システムは、分岐予測に基づく投機的実行を可能にするために、前進ポインタ・スタック履歴ファイル（以下では「APS履歴ファイル」と記す）を具備する。APS履歴ファイルの各々のエントリには、APSの全エントリ及びPP_OFAPSの内容が書き込めるようになっている。

10 (D) 統合レジスタ・ファイル (CRF ; Consolidated Register File)

統合レジスタ・ファイル（以下ではCRFで示す）は、従来のスタックマシンにおけるワード・スタックの内容を、順序不同で保持するものである。

15 第3図は、本第1実施例における、CRF6の各々のエントリ6(i)の詳細な構成を示す説明図である。ここで、iはエントリのアドレスである。CRF6の各々のエントリ6(i)はデータ・フィールド61(i)、書き込み完了フラグ (WCF, Write Completion Flag) フィールド62(i)、カラー (C, Colour) フィールド63(i)及びビジービット (BB) フィールド64(i)から成っている。

20 実際のCRFのハードウェア上の構成は、上述の各フィールド別に設けられたレジスタ・ファイルの集合体である。

CRFの各々のエントリのデータ・フィールドは、1語分のデータが書き込まれる構成となっている。

25 CRFの各々のエントリにおいて、WCFフィールドは、データ・フィールドにデータの書き込みが完了していれば1、完了していなければ0が書き込まれているようになっている。

CRFの各々のエントリにおいて、Cフィールドは、そのCRFのエントリが、命令に含まれるプッシュ操作に対応して割り付けられたものであるのか、アンダーフロー回避のためのデータ・キャッシュからのコード(Fill)の際に割り付けられたものであるのかの区別、前者の場合にはさらに分岐タグが書き込まれるようになっている。本実施例においては、後述するように、分岐タグはAPS履歴ファイルのエントリのアドレスと一定の関係にある。

CRFの各々のエントリにおいて、BBフィールドは、そのCRFのエントリがデータを保持すべく割り付けられている状態であれば1、割り付けられていない状態であれば0が書き込まれているようになっている。

(E) フリー・リスト (FL)

フリー・リスト(以下ではFLで示す)は、フリーな、即ち、割り付けられていない(BBフィールドが0である)CRFのエントリのアドレスを保持するためのメモリであり、本実施例においては、循環型のFIFOキューの構成となっている。

初期状態においては、CRFの全てのエントリのアドレスがFLに登録されている。CRFのフリーなエントリを割り付ける必要がある場合に、FLからフリーなCRFのエントリのアドレスが取り出される。逆に、CRFのあるエントリの割り付けが解除されれば、そのエントリのアドレスがFLに登録されるようになっている。

(F) 命令キュー (IQ; Instruction Queue)

命令キュー(以下ではIQで示す)は、実行または完了を待っているデコード・設定済の命令を保持するメモリであり、循環型のFIFOキューの構成となっている。

第4図は、IQの構成を示す説明図である。第4図において、IQ5

の各々のエントリは下から順に0、1、2、…とアドレスが付けられているものとし、縦線が施されているIQ5のエントリは、実行または完了を待っている命令を保持しているものとする。IQは、設定ポインタ／完了ポインタと名付けた二つのレジスタを具備する。設定ポインタは、次にデコード・設定される命令の内容を書き込むべきエントリを示す。完了ポインタは、次に完了されるべき命令のエントリを示す。完了ポインタの値から設定ポインタの値を引くことで、IQに何エントリの空きがあるかがわかる。初期状態においては、設定ポインタ及び完了ポインタの値は共に0となっている。

第5図は、IQ5の各々のエントリ5(i)の詳細な構成を示す説明図である。ここで、iはエントリのアドレスである。IQ5の各々のエントリ5(i)はオペレーション・フィールド500(i)、オペランド・フィールド501(i)、第1ソース・フィールド510(i)、第1書き込み完了フラグ(WCF1)フィールド511(i)、第2ソース・フィールド520(i)、第2書き込み完了フラグ(WCF2)フィールド521(i)、第3ソース・フィールド530(i)、第3書き込み完了フラグ(WCF3)フィールド531(i)、第4ソース・フィールド540(i)、第4書き込み完了フラグ(WCF4)フィールド541(i)、第1デスティネーション・フィールド55(i)、第2デスティネーション・フィールド56(i)、分岐タグ(BT)フィールド57(i)、及び実行状態(S;State)フィールド58(i)から成っている。

IQの各々のエントリのオペレーション・フィールドはオペレーション・コードが書き込まれる構成となっている。

IQの各々のエントリのオペランド・フィールドは、オペレーション・コードに続いてオペランドが示されるような命令の場合に、このオペラ

ンドが書き込まれるようになっている。

I Qの各々のエントリの第1～第4ソース・フィールドの各々は、ソース・データを保持すべく割り付けられているC R Fのエントリのアドレスが書き込まれるようになっている。オペランド・スタックに対するポップ操作を含む命令の場合には、命令によりポップされるべきデータを保持すべく割り付けられているC R Fのエントリのアドレスが、ポップされる順に書き込まれるようになっている。

I Qの各々のエントリの第1～第2デスティネーション・フィールドの各々は、命令のデコード・設定に伴い、新たに割り付けられるC R Fのエントリのアドレスが書き込まれるようになっている。オペランド・スタックに対するプッシュ操作を含む命令の場合には、命令によりプッシュされるべきデータを保持すべく割り付けられるC R Fのエントリのアドレスが、プッシュされる順に書き込まれるようになっている。

I Qの各々のエントリにおいて、第1～第4の各W C Fフィールドは各々第1～第4ソース・フィールドに対応して設けられている。W C F 15 1フィールドは第1ソース・フィールドに示されるC R Fのエントリにデータの書き込みが完了していれば1、完了していなければ0が書き込まれているようになっている。第2～第4のW C Fフィールド、ソース・フィールドに関しても同様である。

I Qは、各エントリのソース・フィールドごとに比較回路を備えており、データの書き込みが行われるC R Fのエントリのアドレスを各ソース・フィールドの内容と比較して、一致するソース・フィールドに対応するW C Fフィールドに1を立てるような機能を有する。

I Qの各々のエントリのB Tフィールドは、分岐予測に基づく投機的実行に係るもので、本実施例においては、後述するように、B Tフィールドに書き込まれる分岐タグはA P S履歴ファイルのエントリのアドレスと一定の関係にある。

I Q の各々のエントリにおいて、S フィールドは、そのエントリに書き込まれている命令の実行状態に応じて、未実行、実行済み、正常終了、例外事象発生等の情報が書き込まれているようになっている。

(G) 演算ユニット

5 本実施例の計算機システムは、演算ユニット 0 及び演算ユニット 1 を具備しており、その各々は、I Q より送られてくる算術論理演算、データ・タイプの変換演算、比較演算等の演算命令を実行する機能を有し、互いに独立に並行して動作するようになっている。

本発明の計算機システムにおいては、各々の演算ユニットをパイプラ
10 イン化したり、より多くの演算ユニットを具備したり、演算ユニットごとに実行する演算の種類を特定した構成とすることも可能である。

(H) 分岐ユニット

分岐ユニットは、I Q より送られてくる条件分岐命令を処理し、分岐の有無を確定して、命令フェッチ・ユニットに通知する機能を有する。

15 (I) ロード/ストア・ユニット (L S U ; Load/Store Unit)

ロード/ストア・ユニット (以下では L S U で示す) は、アドレス計算を行う機能を有し、データ・キャッシュ及び C R F にアクセスすることができるようになっている。

L S U は、最初のローカル変数へのポインタを保持する図示してないレジスタ (vars レジスタ) を具備する。本実施例の計算機システムにおいては、最初のローカル変数の格納域はデータ・キャッシュあるいは C R F にあるが、vars レジスタには、データ・キャッシュにおける相当するアドレス値が書き込まれているようになっている。すなわち、全てあるいは一部のローカル変数の実際の格納域が C R F にある場合でも、各々のローカル変数に、全てのローカル変数をデータ・キャッシュにストア (Spill) したと仮定した場合のデータ・キャッシュにおけるアドレ

ス値を対応させることができるので、ロード／ストア命令の処理において、LSUは vars レジスタの値を用いてアドレス計算を行い、対象となるローカル変数の格納域がデータ・キャッシュかCRFかを判定し、その格納域にアクセスする。格納域がCRFにあると判定された場合には、アクセスすべきCRFのエントリのアドレスはAPSから読み出される。

ロード／ストア命令が命令デコード・設定ユニットでデコードされると、その内容が、命令キューに書き込まれると同時に、LSUにも送られるようになっている。

LSUは、プログラム上の順番で、ロード／ストア命令を命令キューのエントリのアドレス、対象となる変数名及びデータと共に蓄える図示してないキューを具備する。このキューは連想機能を備えており、変数名を照合してデータ・アクセスの依存性の検証を行うことにより、ロード命令の実行を out-of-order で行うことができるようになっている。本実施例の計算機システムにおいては、ローカル変数の格納域はデータ・キャッシュあるいはCRFにあるが、上記キューにすでに同じ変数名の書き込みがあるローカル変数のロード命令の場合、変数データは上記キューより読み出される。ストア命令の実行は、正確な例外処理を保証するために、後述するように、完了ステージにおいて in-order で行われる。

LSUは、プログラム中に示されるロード／ストア命令を実行すると共に、オーバーフロー／アンダーフローの回避のため、APS、CPS及びCRFの空きに応じて、CRFに保持されているスタックの最下位にあたるデータをデータ・キャッシュとの間で自動的にストア(Spill)／ロード(Fill)するようになっている。

本発明第1実施例の計算機システムにおいては、APS、CPS及びCRFのオーバーフロー／アンダーフローは以下に示すようなメカニ

ズムにより回避される。

APS、CPSあるいはCRFの空きが一定量以下になると、オーバーフローを回避するために以下のような制御動作が隨時行われるようになっている。

5 すなわち、APSがCRFのエントリのアドレスを一定量以上保持していない場合、あるいは、BP_OF_PS で示されるエントリの内容がAPSとCPSで一致しない場合、以上の条件が解消するまでの間、命令シーケンスの流れを停止するよう命令フェッチ・ユニットに信号が送られる。

10 逆に、APSがCRFのエントリのアドレスを一定量以上保持しており、BP_OF_PS で示されるエントリの内容がAPSとCPSで一致する場合には、その2つのポインタ・スタックで一致する BP_OF_PS で示されるエントリの内容で示されるCRFのエントリに書き込まれている1語分のデータをデータ・キャッシュにストア (Spill) し、BP_OF_PS の値に1を加える。さらに、上記CRFのエントリのBBフィールドを0 15 に変更し、そのエントリのアドレスをFLに登録する。

APS、CPS及びCRFの空きがいずれも一定量以上になると、アンダーフローを回避するために、最後にストア (Spill) した1語分のデータをデータ・キャッシュから取り出し、それにFLに登録されている20 フリーなCRFの1エントリを割り付け、そのデータ・フィールドに書き込む。WCF、BBの各フィールドは1とする。さらに、その割り付けられたCRFのエントリのアドレスを、APS及びCPSのBP_OF_PS で示されるエントリの1つ下に各々書き込み、BP_OF_PS の値から1を引く。

25 また、CRFとデータ・キャッシュの間の Spill/Fill の動作を効率的に行うために、LSUがデータをいったん蓄えるバッファを備え、この中

に適當な語数のデータを溜めておくような構成とすることも可能である。

ついで、本発明第1実施例の計算機システムの動作を説明する。

本実施例の計算機システムは命令を、①命令フェッチ、②命令デコード・設定、③実行、④完了の4ステージで処理する。当分の間、説明を簡単にするため、1サイクルで1つの命令をデコード・設定／完了できるものとして、以下に各ステージごとに動作内容を説明する。

① 命令フェッチ・ステージ

このステージでは、命令フェッチ・ユニットが命令キャッシュから命令を取り出すと共に、次にフェッチする命令のアドレスを決定する。次に命令をフェッチするのは通常次アドレス値からであるが、フェッチした命令が無条件分岐命令であるか、条件分岐命令で分岐すると予測した場合、分岐予測が外れた場合、あるいは例外が発生した場合には、フェッチするアドレス値を変更する。

② 命令デコード・設定ステージ

このステージでは、命令をデコードして、命令の内容に応じて前進ポインタ・スタック (APS) 及び統合レジスタ・ファイル (CRF) を操作すると共に命令の内容を命令キュー (IQ) に書き込むことにより、命令に含まれる演算等がデータ駆動で実行されるべく設定する。以下に、設定動作を詳細に説明する。

本発明の計算機システムにおいては、従来のスタック・マシンにおけるワード・スタックのスタックトップ近傍がポインタ・スタックとCRFによって再現されるが、命令に含まれるオペランド・スタックに対するスタック操作が、APSに対して同様に適用される。

1語のデータのオペランド・スタックへのプッシュ操作をエミュレートするには、FLに登録されているフリーなCRFの1エントリをその

データを保持すべく割り付け、そのエントリのアドレスをAPSにプッシュすればよい。

オペランド・スタックの操作命令 (Java VM における pop, pop2, dup, dup2, dup_x1, dup2_x1, dup_x2, dup2_x2, swap) の場合、基本的には、オペランド・スタックに対して行うべき操作をAPSに対して同様に行えばよい。本第1実施例においては、スタック上でコピーを作成するようなオペランド・スタックの操作命令 (Java VM における dup, dup2, dup_x1, dup2_x1, dup_x2, dup2_x2) の場合には、コピー・データを保持すべくFLに登録されているフリーなCRFのエントリを割り付け、そのエントリのアドレスをAPSの適切なエントリに書き込むようになっている。

命令のデコード・設定に伴い新たに割り付けられるCRFのエントリにおいては、BBフィールドに1を立て、Cフィールドには命令デコード・設定ユニットから送られてくる分岐タグを書き込む。即値データのプッシュ命令の場合には、データがすでに得られているので、データ・フィールドにそのデータを書き込み、WCFフィールドに1を立てる。それ以外の場合には、データはデコード・設定の時点では得られていないので、WCFフィールドを0としておく。

デコードされた命令の内容をプログラム上の順番でIQに保持しておくために、その命令の内容をIQの設定ポインタで示されるエントリに書き込み、設定ポインタの値に1を加える。すなわち、オペレーション・フィールドにオペレーション・コードを書き込み、オペレーション・コードに続いてオペランドが示されるような命令の場合には、オペランド・フィールドにこのオペランドを書き込む。BTフィールドには命令デコード・設定ユニットから送られてくる分岐タグを書き込む。Sフィールドに関しては、無条件分岐命令、即値データのオペランド・スタック

へのプッシュ命令あるいはスタック上でコピーを作成することのないオペランド・スタックの操作命令 (Java VM における `pop`, `pop2`, `swap`) の場合は実行済みとし、その他の命令の場合は未実行としておく。

オペランド・スタックに対するポップ操作を含む命令の場合には、ポップすべき語数と同じ数だけAPSからポップされるCRFのエントリのアドレスを、その順で第1～第4ソース・フィールドに書き込む。この際、エントリ・アドレスがポップされるCRFのエントリの各々でWCFFフィールドを読み出し、IQの対応するWCFFフィールドに送る。

オペランド・スタックに対するプッシュ操作を含む命令の場合には、プッシュすべき語数と同じ数だけAPSにプッシュされるCRFのエントリのアドレスを、その順で第1～第2デスティネーション・フィールドに書き込む。

本第1実施例においては、スタック上でコピーを作成するようなオペランド・スタックの操作命令の場合には、コピー元となるデータを保持すべく割り付けられているCRFのエントリのアドレスをソース・フィールドに、コピー・データを保持すべく新たに割り付けられるCRFのエントリのアドレスをデスティネーション・フィールドに、一定の対応関係のもとに書き込む。

命令の種類に応じて、オペランド・スタックに対してポップ/プッシュすべき語数 (オペランド・スタックの操作命令の場合には、作成するコピーの語数) は決まっているので、オペレーション・フィールドの内容によって、第1～第4ソース・フィールド、第1～第4のWCFFフィールド及び第1～第2デスティネーション・フィールドのうちのいずれが有効であるかを知ることができる。

ロード/ストア命令の場合には、その内容を、IQに書き込むと同時に、その書き込みが行われるIQのエントリのアドレスと共にLSUに

送る。

③ 実行ステージ

I Qに保持されている未実行の命令は、データ駆動で処理される。従って、命令実行順序は *out-of-order* になる。以下に、命令の種類ごとに、
5 実行ステージにおける動作を説明する。

(a) 即値データのオペランド・スタックへのプッシュ命令

実行ステージにおける動作としては、何も行わない。

(b) 変数データのオペランド・スタックへのロード命令

I Qの、ロード命令を書き込みの内容とするエントリに関しては、同じ内容が命令デコード・設定ステージにおいて L S U に送られている。
10 L S U では、送られてきたロード命令を *out-of-order* で処理するようになっている。

I Qにおいて、オペランド・スタック上のデータをポップしてアドレス計算を行うようなロード命令 (Java VM における *iaload*, *laload*, *faload*,
15 *daload*, *aaload*, *baload*, *caload*, *saload*) を書き込みの内容とし、有効なソース・フィールドに対応する W C F フィールドが全て 1 となっている (ソース・データが全て C R F に書き込み済みとなっている) エントリがあれば、そのエントリのアドレスと共に C R F をアクセスして得られるソース・データを L S U に送る。L S U は送られてきたソース・データをもとにアドレス計算を行い、ロードの実行を試みる。

L S U で変数データが得られると、デスティネーションである C R F のエントリのデータ・フィールドに変数データを書き込み、W C F フィールドを 1 に変更する。その上、I Qにおいて各ソース・フィールドで上記デスティネーションである C R F のエントリのアドレスを照合し、
25 一致するソース・フィールドに対応する W C F フィールドを 1 とする。この際、同じタイミングで命令が書き込まれる I Q のエントリにおいて

は、その書き込まれる内容と比較するようになっている。以上の動作が正常に終了すれば、そのロード命令を保持している I Q のエントリの S フィールドを正常終了に変更する。

(c) オペランド・スタック上のデータの変数へのストア命令

5 I Qにおいて、ストア命令を書き込みの内容とし、有効なソース・フィールドに対応する WCF フィールドが全て 1 となっている（ソース・データが全て C R F に書き込み済みとなっている）エントリがあれば、そのエントリのアドレスと共に C R F をアクセスして得られるソース・データを L S U に送る。

10 オペランド・スタック上のデータをポップしてアドレス計算を行うようなストア命令（Java VM における iastore, lastore, fastore, dastore, aastore, bastore, castore, sastore）の場合、L S U は送られてきたソース・データをもとにアドレス計算を行う。

15 以上の動作が正常に終了すれば、そのストア命令を保持している I Q のエントリの S フィールドをストア実行可能に変更する。

正確な例外処理を保証するために、実際のストアの実行は完了ステージにおいて行う。

(d) 演算命令

20 I Qにおいて、演算命令を書き込みの内容とし、有効なソース・フィールドに対応する WCF フィールドが全て 1 となっている（ソース・データが全て C R F に書き込み済みとなっている）エントリがあり、利用可能な状態の演算ユニットがあれば、そのエントリの内容をそのエントリのアドレス及び C R F をアクセスして得られるソース・データと共に利用可能な演算ユニットに送り実行させる。

25 演算の実行が正常に終了すれば、デスティネーションである C R F のエントリのデータ・フィールドに演算結果を書き込み、WCF フィール

ドを 1 に変更する。その上、I QにおけるC R F のエントリのアドレスの照合及びW C F フィールドの変更を、上述のロード命令の場合と同様に行う。以上の動作が正常に終了すれば、その演算命令を保持している I QのエントリのS フィールドを正常終了に変更する。

5 (e) オペランド・スタックの操作命令

スタック上でコピーを作成することのないオペランド・スタックの操作命令に関しては、実行ステージにおける動作としては、何も行わない。

本第 1 実施例においては、I Qにおいて、スタック上でコピーを作成するようなオペランド・スタックの操作命令を書き込みの内容とし、有効なソース・フィールドに対応するW C F フィールドが全て 1 となっている（ソース・データが全てC R F に書き込み済みとなっている）エントリがあれば、その有効なソース・フィールドに示されるC R F のエントリからデータを読み出し、これを対応するデスティネーション・フィールドに示されるC R F のエントリのデータ・フィールドに書き込み、W C F フィールドを 1 に変更する。その上、I QにおけるC R F のエントリのアドレスの照合及びW C F フィールドの変更を、上述のロード命令の場合と同様に行う。以上の動作が正常に終了すれば、そのオペランド・スタックの操作命令を保持している I QのエントリのS フィールドを正常終了に変更する。

20 (f) 分岐命令

無条件分岐命令に関しては、実行ステージにおける動作としては、何も行わない。

I Qにおいて、条件分岐命令を書き込みの内容とし、有効なソース・フィールドに対応するW C F フィールドが全て 1 となっている（ソース・データが全てC R F に書き込み済みとなっている）エントリがあれば、そのエントリの内容をそのエントリのアドレス及びC R F をアクセスし

て得られるソース・データと共に分岐ユニットに送り実行させる。

分岐ユニットにおいて条件式の計算が正常に終了すれば、その結果を分岐先アドレスと共に命令フェッチ・ユニットに通知する。以上の動作が正常に終了すれば、その条件分岐命令を保持している I Q のエントリの S フィールドを正常終了に変更する。

以上のように、I Q に保持されている未実行の命令は、実行可能となったものから処理されるので、命令実行順序は *out-of-order* になる。また、演算ユニット 0, 1、分岐ユニット及びロード／ストア・ユニットの各実行ユニットは互いに独立に並行して動作する。

ある命令の処理において例外事象が発生した場合には、その情報を、その命令を保持している I Q のエントリの S フィールドに書き込むと共に、命令フェッチ・ユニットに例外ベクタを通知する。

④ 完了ステージ

ある命令が完了できるためには、プログラム上の順番でその命令よりも前にある命令が全て完了していなくてはならない。

I Q の完了ポインタで示されるエントリにおいて、S フィールドが実行済み／正常終了である、あるいはそうなると、そのエントリに書き込まれている命令の内容に基づいて C P S 及び C R F を操作し、完了ポインタの値に 1 を加える。

C P S は、命令がデコード・設定された際の A P S の動作を再現すべく操作される。すなわち、ポップ／プッシュ操作を含む命令の場合には、有効なソース・フィールドの内容と同じものを順に C P S からポップし、有効なデスティネーション・フィールドの内容を順に C P S にプッシュする。スタック上でコピーを作成することのないオペランド・スタックの操作命令の場合には、オペランド・スタックに対して行うべき操作を C P S に対して全く同様に行えばよい。本第 1 実施例においては、スタ

ック上でコピーを作成するようなオペランド・スタックの操作命令の場合には、有効なソース・フィールド及び有効なデスティネーション・フィールドを参照して、その命令のデコード・設定の際にAPSに対して行われた操作がCPSにおいて再現される。

5 本第1実施例においては、上述のCPSに対する操作に伴い、エントリ・アドレスがCPSからポップされるCRFのエントリでは、BBフィールドを0に変更し、そのエントリ・アドレスをFLに登録する。

IQの完了ポインタで示されるエントリにおいて、ストア命令が書き込まれている場合には、Sフィールドがストア実行可能である、あるいはそうなると、LSUに、上記ストア命令を保持しているIQのエントリのアドレスを示して、実際のストアの実行を依頼する。こうすれば、データが in-order でストアされることが保証できる。さらに、CPS及びCRFに対する操作を上と同様に行い、完了ポインタの値に1を加える。

15 以上のように、完了ポインタの値に1が加えられることによって、キューから除外されたIQのエントリに保持されていた命令は、完了したことになる。その命令よりも前にデコード・設定された命令はすべて完了しているので、命令の完了は in-order で行われることになる。

IQの完了ポインタで示されるエントリにおいて、Sフィールドが例外事象発生である、あるいはそうなった場合には、その時点におけるCPS及びCRFによって、プログラムが in-order で実行された場合の例外発生時点の状態が構成されるので、正確な例外処理が可能である。例外事象の発生した命令以降にデコード・設定された命令を全てキャンセルするには、キャンセルされるべき命令が書き込まれているIQのエントリの有効なデスティネーション・フィールドに示されるCRFのエントリの各々に対して、そのBBフィールドを0に戻し、そのエントリ・

アドレスを F L に登録することによって、割り付けを解除し、完了ポインタの値に 1 を加えたものを設定ポインタに書き込むことによって、キャンセルされるべき命令を保持している I Q のエントリを全てキューから除外すればよい。

5 以上が、本発明第 1 実施例の計算機システムの動作についての全般的な説明であるが、ついで、具体的な動作例について説明する。

いま、本第 1 実施例の計算機システムで、以下のようなプログラムを実行することを考えよう。

dload [A] (変数名 [A] に対応する倍精度浮動小数点データのロード)
10 dload [B] (変数名 [B] に対応する倍精度浮動小数点データのロード)
dadd (倍精度浮動小数点データ間の加算)
d2f (倍精度浮動小数点データの単精度浮動小数点データへの変換)
fload [T] (変数名 [T] に対応する単精度浮動小数点データのロード)
15 dup_x1 (スタックトップの語のコピーを作成し、先頭から 3 語目に割り込ませる)
fdiv (単精度浮動小数点データ間の除算)
fsub (単精度浮動小数点データ間の減算)
fstore [X] (スタックトップにある単精度浮動小数点データの変数名
20 [X] に対応する格納域へのストア)

以上のプログラムは、 $X = T - (A + B) / T$ の計算を行うものであるが、A と B のデータが倍精度で与えられ、この間の加算を倍精度のまま実行して、得られた加算データを単精度に変換して、以降は単精度で計算を行う、というものである。

25 第 6 図～第 12 図は、本第 1 実施例の計算機システムにおいて、上記プログラムを処理する際の動作をサイクル毎に示した説明図であり、以

下ではこの図をもとに詳細な動作を説明する。第6図～第12図において、CRF6及びIQ5の各エントリの構成は、それぞれ第3図、第5図のものと同じである。第6図～第12図で空白となっている箇所は、そのフィールドの内容に留意する必要が無いことを意味する。時系列で各構成要素の内容を示すために、各部の符号の後尾にハイフンと各サイクルに対応する数字を添えている。また、第6図～第12図において、APS、CPS、IQ及びCRFの各エントリは下から順に0、1、2、～のようにアドレスが付けられているものとする。

本動作例においては、説明を簡単にするため、変数データは全てデータ・キャッシュに格納され、CRFとデータ・キャッシュの間のSpill/Fillの動作は行わないものとする。従って、BP_OF_PSの値は終始0である。

また、本動作例においては、当初、APS、CPS、IQ及びCRFは初期化されており、FLにCRFの全てのエントリのアドレスが順に<0>、<1>、<2>、<3>…と書き込まれていて、この順で取り出されるものとする。

以下に、各サイクルにおける動作を、(A) 命令デコード・設定、(B) 実行及び(C) 完了の各ステージに分けて詳細に説明する。

(1-A) 第1サイクルの命令デコード・設定ステージ

命令 `dload [A]` のデコード・設定を行う。倍長語の変数データのオペランド・スタックへのロード命令であるので、FLに登録されているフリーなCRFの2エントリ6(0)、6(1)をそのデータを保持すべく割り付け、そのエントリのアドレス<0>、<1>をAPSにプッシュし、APSは3-1のようになる。

CRFの6(0)、6(1)の各エントリにおいては、BBフィールドに1を立て、WCF及びCの各フィールドには0を書き込み、CRFは6-1のようになる。ここで、本動作例においては、終始分岐タグとして命

令デコード・設定ユニットから0が送られてくるものとする。

5 設定ポインタの値は0であるので、IQのエントリ5(0)に上記命令の内容を書き込み、IQは5-1のようになる。この際、APSにプッシュされるCRFのエントリのアドレス〈0〉,〈1〉を各々第1、第2デスティネーション・フィールドに書き込んでいる。さらに、設定ポインタの値に1を加え1にする。ここで、本動作例においては、IQのSフィールドには、命令が未実行であれば0、実行済み／正常終了あるいはストア命令におけるストア実行可能であれば1が書き込まれるものとする。

10 IQのエントリ5(0)に書き込まれるものと同じ上記命令の内容を、IQのエントリのアドレス0と共にLSUに送る。

(1-B) 第1サイクルの実行ステージ

当初のIQにおいては、実行可能な命令が書き込まれているエントリは存在しないので、実行ステージの動作としては何も行われない。

15 (1-C) 第1サイクルの完了ステージ

当初のIQの完了ポインタが示すエントリ5(0)において、命令はまだ書き込まれていないため、完了ステージの動作としては何も行われない。

(2-A) 第2サイクルの命令デコード・設定ステージ

20 命令dload [B]のデコード・設定を行う。倍長語の変数データのオペランド・スタックへのロード命令であるので、FLに登録されているフリーなCRFの2エントリ6(2)、6(3)をそのデータを保持すべく割り付け、そのエントリのアドレス〈2〉,〈3〉をAPSにプッシュし、APSは3-2のようになる。

25 CRFの6(2)、6(3)の各エントリにおいては、BBフィールドに1を立て、WC F及びCの各フィールドには0を書き込み、CRFは6-

2のようになる。

設定ポインタの値は1であるので、IQのエントリ5(1)に上記命令の内容を書き込み、IQは5-2のようになる。この際、APSにプッシュされるCRFのエントリのアドレス〈2〉, 〈3〉を各々第1、第5 2デスティネーション・フィールドに書き込んでいる。さらに、設定ポインタの値に1を加え2にする。

IQのエントリ5(1)に書き込まれるものと同じ上記命令の内容を、IQのエントリのアドレス1と共にLSUに送る。

(2-B) 第2サイクルの実行ステージ

10 LSUは変数Aのアクセスを開始する。レイテンシは2サイクルであるとする。

(2-C) 第2サイクルの完了ステージ

5-1の状態にあるIQの完了ポインタが示すエントリ5(0)において、Sフィールドは0であるので、完了ステージの動作としては何も行15 われない。

(3-A) 第3サイクルの命令デコード・設定ステージ

命令daddのデコード・設定を行う。オペランド・スタックから4語のソース・データをポップして演算を行い、倍長語の演算結果をプッシュする演算命令であるので、APSから〈0〉, 〈1〉, 〈2〉, 〈3〉をポップし、FLに登録されているフリーなCRFの2エントリ6(4)、6(5)を演算結果を保持すべく割り付け、そのエントリのアドレス〈4〉, 〈5〉をAPSにプッシュし、APSは3-3のようになる。

CRFの6(4)、6(5)の各エントリにおいては、BBフィールドに1を立て、WCF及びCの各フィールドには0を書き込む。

25 設定ポインタの値は2であるので、IQのエントリ5(2)に上記命令の内容を書き込む。この際、APSからポップされるCRFのエントリ

のアドレス〈0〉, 〈1〉, 〈2〉, 〈3〉を各々第1～第4ソース・フィールドに、APSにプッシュされる〈4〉, 〈5〉を各々第1、第2デスティネーション・フィールドに書き込んでいる。また、6-2の状態にあるCRFの6(0)、6(1)、6(2)、6(3)の各エントリでWCFフィールドを読み出し、それぞれIQのWCF1～4の各フィールドに送っている。さらに、設定ポインタの値に1を加え3にする。

(3-B) 第3サイクルの実行ステージ

LSUは変数Bのアクセスを開始する。レイテンシは2サイクルであるとする。

LSUから変数Aのデータを構成する2語A_1、A_2が送られてくるので、CRFのエントリ6(0)、6(1)のデータ・フィールドに各々書き込み、WCFフィールドを1に変更する。その上、IQにおいて各ソース・フィールドでCRFのエントリのアドレス〈0〉, 〈1〉を照合するが、この場合、同じタイミングで命令が書き込まれるIQのエントリ5(2)の第1、第2ソース・フィールドで一致するので、同じエントリのWCF1、2の各フィールドを1とする。(WCF3、4の各フィールドについては、(3-A)で説明したように6-2の状態にあるCRFの対応するエントリのWCFフィールドが読み出され送られてくるので、これが書き込まれる。)

以上のようにIQのエントリ5(0)に書き込まれた命令の実行が正常に終了するので、5(0)のSフィールドを正常終了を意味する1に変更する。

(3-C) 第3サイクルの完了ステージ

5-2の状態にあるIQの完了ポインタが示すエントリ5(0)において、Sフィールドは0であるので、完了ステージの動作としては何も行われない。

(4-A) 第4サイクルの命令デコード・設定ステージ

命令 d2f のデコード・設定を行う。オペランド・スタックから 2 語のソース・データをポップして変換演算を行い、1 語の演算結果をプッシュする演算命令であるので、APS から (4), (5) をポップし、FL に登録されているフリーな CRF のエントリ 6 (6) を演算結果を保持すべく割り付け、そのエントリのアドレス (6) をAPS にプッシュし、APS は 3-4 のようになる。

CRF のエントリ 6 (6)においては、BB フィールドに 1 を立て、WC F 及び C の各フィールドには 0 を書き込む。

設定ポインタの値は 3 であるので、IQ のエントリ 5 (3) に上記命令の内容を書き込む。この際、APS からポップされる CRF のエントリのアドレス (4), (5) を各々第 1、第 2 ソース・フィールドに、APS にプッシュされる (6) を第 1 デスティネーション・フィールドに書き込んでいる。また、6-3 の状態にある CRF の 6 (4), 6 (5) の各エントリで WCF フィールドを読み出し、それぞれ IQ の WCF 1, 2 の各フィールドに送っている。さらに、設定ポインタの値に 1 を加え 4 にする。

(4-B) 第4サイクルの実行ステージ

LSU から変数 B のデータを構成する 2 語 B_1, B_2 が送られてくるので、CRF のエントリ 6 (2), 6 (3) のデータ・フィールドに各々書き込み、WCF フィールドを 1 に変更する。その上、IQ において各ソース・フィールドで CRF のエントリのアドレス (2), (3) を照合するが、この場合、IQ のエントリ 5 (2) の第 3、第 4 ソース・フィールドで一致するので、同じエントリの WCF 3, 4 の各フィールドを 1 とする。

以上のように IQ のエントリ 5 (1) に書き込まれた命令の実行が正常に終了するので、5 (1) の S フィールドを正常終了を意味する 1 に変更

する。

(4-C) 第4サイクルの完了ステージ

5-3の状態にあるIQの完了ポインタが示すエントリ5(0)において、Sフィールドが1となったので、5(0)の内容に基づいてCPS(及びCRF)を操作する。すなわち、IQのエントリ5(0)のデスティネーション・フィールドに書き込まれている〈0〉, 〈1〉をCPSにプッシュし、CPSは4-4のようになる。さらに、完了ポインタの値に1を加え1とし、これで、5(0)の命令は完了したことになる。

(5-A) 第5サイクルの命令デコード・設定ステージ

命令 float [T] のデコード・設定を行う。1語の変数データのオペランド・スタックへのロード命令であるので、FLに登録されているフリーなCRFのエントリ6(7)をそのデータを保持すべく割り付け、そのエントリのアドレス〈7〉をAPSにプッシュし、APSは3-5のようになる。

CRFのエントリ6(7)においては、BBフィールドに1を立て、WC及びCの各フィールドには0を書き込む。

設定ポインタの値は4であるので、IQのエントリ5(4)に上記命令の内容を書き込む。この際、APSにプッシュされるCRFのエントリのアドレス〈7〉を第1デスティネーション・フィールドに書き込んでいる。さらに、設定ポインタの値に1を加え5にする。

IQのエントリ5(4)に書き込まれるものと同じ上記命令の内容を、IQのエントリのアドレス4と共にLSUに送る。

(5-B) 第5サイクルの実行ステージ

5-4の状態にあるIQにおいて、演算命令が書き込まれているエントリ5(2)はWC Fフィールドが全て1となっているので、このエントリ5(2)の内容をそのエントリのアドレス2及びCRFの6(0)、6(1)、

6 (2)、6 (3)の各エントリに書き込まれているソース・データと共に演算ユニット0に送り演算を開始させる。この演算のレイテンシは2サイクルであるとする。

(5-C) 第5サイクルの完了ステージ

5 5-4の状態にあるIQの完了ポインタが示すエントリ5(1)において、Sフィールドが1となったので、5(1)の内容に基づいてCPS(及びCRF)を操作する。すなわち、IQのエントリ5(1)のデスティネーション・フィールドに書き込まれている〈2〉、〈3〉をCPSにプッシュし、CPSは4-5のようになる。さらに、完了ポインタの値に1を加え2とし、これで、5(1)の命令は完了したことになる。

10 (6-A) 第6サイクルの命令デコード・設定ステージ

15 命令dup_x1のデコード・設定を行う。命令dup_x1は、ワード・スタックが、(右方向に成長するものとして) ..., word1, word2のような状態であるとき、これを ..., word2, word1, word2と変えるような、スタック上で1語のコピーを作成するオペランド・スタックの操作命令であるので、FLに登録されているフリーなCRFのエントリ6(8)をコピー・データを保持すべく割り付け、3-5のように下から〈6〉、〈7〉となっている状態のAPSを3-6のように〈8〉、〈6〉、〈7〉と変える。

20 CRFのエントリ6(8)においては、BBフィールドに1を立て、WC及びCの各フィールドには0を書き込む。

25 設定ポインタの値は5であるので、IQのエントリ5(5)に上記命令の内容を書き込む。この際、コピー元となるデータを保持すべく割り付けられているCRFのエントリのアドレス〈7〉を第1ソース・フィールドに、コピー・データを保持すべく新たに割り付けられるCRFのエントリのアドレス〈8〉を第1デスティネーション・フィールドに書き込んでいる。また、6-5の状態にあるCRFのエントリ6(7)でWC

F フィールドを読み出し、 I Q の W C F 1 フィールドに送っている。さらに、設定ポインタの値に 1 を加え 6 にする。

(6-B) 第 6 サイクルの実行ステージ

L S U は変数 T のアクセスを開始する。レイテンシは 2 サイクルであるとする。

演算ユニット 0 で 5 (2) の演算命令の実行が正常に終了すれば、演算結果を構成する 2 語 (A+B)_1, (A+B)_2 が送られてくるので、 C R F のエントリ 6 (4)、 6 (5) のデータ・フィールドに各々書き込み、 W C F フィールドを 1 に変更する。その上、 I Q において各ソース・フィールドで C R F のエントリのアドレス <4>, <5> を照合するが、この場合、 I Q のエントリ 5 (3) の第 1 、第 2 ソース・フィールドで一致するので、同じエントリの W C F 1 、 2 の各フィールドを 1 とする。

以上のように I Q のエントリ 5 (2) に書き込まれた命令の実行が正常に終了するので、 5 (2) の S フィールドを正常終了を意味する 1 に変更する。

(6-C) 第 6 サイクルの完了ステージ

5-5 の状態にある I Q の完了ポインタが示すエントリ 5 (2) において、 S フィールドは 0 であるので、完了ステージの動作としては何も行われない。

(7-A) 第 7 サイクルの命令デコード・設定ステージ

命令 fdiv のデコード・設定を行う。オペランド・スタックから 2 語のソース・データをポップして演算を行い、 1 語の演算結果をプッシュする演算命令であるので、 A P S から <6>, <7> をポップし、 F L に登録されているフリーな C R F のエントリ 6 (9) を演算結果を保持すべく割り付け、そのエントリのアドレス <9> を A P S にプッシュし、 A P S は 3-7 のようになる。

CRFのエントリ6(9)においては、BBフィールドに1を立て、WC
F及びCの各フィールドには0を書き込む。

設定ポインタの値は6であるので、IQのエントリ5(6)に上記命令
の内容を書き込む。この際、APSからポップされるCRFのエントリ
のアドレス〈6〉、〈7〉を各々第1、第2ソース・フィールドに、AP
Sにプッシュされる〈9〉を第1デスティネーション・フィールドに書
き込んでいる。また、6-6の状態にあるCRFの6(6)、6(7)の各エ
ントリでWCFフィールドを読み出し、それぞれIQのWCF1、2の
各フィールドに送っている。さらに、設定ポインタの値に1を加え7に
する。

(7-B) 第7サイクルの実行ステージ

5-6の状態にあるIQにおいて、演算命令が書き込まれているエン
トリ5(3)は有効なWCFフィールドが全て1となっているので、この
エントリ5(3)の内容をそのエントリのアドレス3及びCRFの6(4)、
6(5)の各エントリに書き込まれているソース・データと共に演算ユニッ
ト0に送り演算を開始させる。この演算のレイテンシは2サイクルであ
るとする。

LSUから変数Tのデータが送られてくるので、CRFのエントリ6
(7)のデータ・フィールドに書き込み、WCFフィールドを1に変更する。
その上、IQにおいて各ソース・フィールドでCRFのエントリのアド
レス〈7〉を照合するが、この場合、IQのエントリ5(5)の第1ソ
ース・フィールド及び同じタイミングで命令が書き込まれる5(6)の第2ソ
ース・フィールドで一致するので、5(5)のWCF1フィールド及び5(6)
のWCF2フィールドを1とする。(5(6)のWCF1フィールドについ
ては、(7-A)で説明したように6-6の状態にあるCRFのエント
リ6(6)のWCFフィールドが読み出され送られてくるので、これが書

き込まれる。)

以上のように I Q のエントリ 5 (4) に書き込まれた命令の実行が正常に終了するので、5 (4) の S フィールドを正常終了を意味する 1 に変更する。

5 (7-C) 第 7 サイクルの完了ステージ

5-6 の状態にある I Q の完了ポインタが示すエントリ 5 (2) において、S フィールドが 1 となったので、5 (2) の内容に基づいて C P S 及び C R F を操作する。すなわち、I Q のエントリ 5 (2) のソース・フィールドに書き込まれている <0>, <1>, <2>, <3> を C P S からポップし、デスティネーション・フィールドに書き込まれている <4>, <5> を C P S にプッシュし、C P S は 4-7 のようになる。エントリ・アドレスが C P S からポップされる C R F の 6 (0), 6 (1), 6 (2), 6 (3) の各エントリでは、B B フィールドを 0 に変更する。C R F のエントリのアドレス <0>, <1>, <2>, <3> を F L に登録する。さらに、完了ポインタの値に 1 を加え 3 とし、これで、5 (2) の命令は完了したことになる。

(8-A) 第 8 サイクルの命令デコード・設定ステージ

命令 fsub のデコード・設定を行う。オペランド・スタックから 2 語のソース・データをポップして演算を行い、1 語の演算結果をプッシュする演算命令であるので、A P S から <8>, <9> をポップし、F L に登録されているフリーな C R F のエントリ 6 (10) を演算結果を保持すべく割り付け、そのエントリのアドレス <10> を A P S にプッシュし、A P S は 3-8 のようになる。

C R F のエントリ 6 (10) においては、B B フィールドに 1 を立て、W C F 及び C の各フィールドには 0 を書き込む。

設定ポインタの値は 7 であるので、I Q のエントリ 5 (7) に上記命令

の内容を書き込む。この際、APSからポップされるCRFのエントリのアドレス〈8〉、〈9〉を各々第1、第2ソース・フィールドに、APSにプッシュされる〈10〉を第1デスティネーション・フィールドに書き込んでいる。また、6-7の状態にあるCRFの6(8)、6(9)の各エントリでWCFフィールドを読み出し、それぞれIQのWCF1、2の各フィールドに送っている。さらに、設定ポインタの値に1を加え8にする。

(8-B) 第8サイクルの実行ステージ

演算ユニット0で5(3)の変換演算命令の実行が正常に終了すれば、
10 1語の演算結果(A+B)が送られてくるので、CRFのエントリ6(6)のデータ・フィールドに書き込み、WCFフィールドを1に変更する。その上、IQにおいて各ソース・フィールドでCRFのエントリのアドレス〈6〉を照合するが、この場合、IQのエントリ5(6)の第1ソース・フィールドで一致するので、同じエントリのWCF1フィールドを1とする。

以上のようにIQのエントリ5(3)に書き込まれた命令の実行が正常に終了するので、5(3)のSフィールドを正常終了を意味する1に変更する。

5-7の状態にあるIQにおいて、スタック上でコピーを作成するようなオペランド・スタックの操作命令が書き込まれているエントリ5(5)は有効なWCFフィールドが1となっているので、データのコピーを実行する。すなわち、CRFのエントリ6(7)からデータを読み出し、これをCRFのエントリ6(8)のデータ・フィールドに書き込み、WCFフィールドを1に変更する。その上、IQにおいて各ソース・フィールドでCRFのエントリのアドレス〈8〉を照合するが、この場合、同じタイミングで命令が書き込まれるIQのエントリ5(7)の第1ソース・フィ

ールドで一致するので、同じエントリのWCF 1 フィールドを1とする。

(WCF 2 フィールドについては、(8-A) で説明したように6-7の状態にあるCRFのエントリ6(9)のWCFフィールドが読み出され送られてくるので、これが書き込まれる。) IQのエントリ5(5)のSフィールドを正常終了を意味する1に変更する。

5 (8-C) 第8サイクルの完了ステージ

5-7の状態にあるIQの完了ポインタが示すエントリ5(3)において、Sフィールドは0であるので、完了ステージの動作としては何も行われない。

10 (9-A) 第9サイクルの命令デコード・設定ステージ

命令 `fstore [X]` のデコード・設定を行う。スタックトップにある1語のデータのストア命令であるので、APSから〈10〉をポップし、APSは3-9のようになる。

15 設定ポインタの値は8であるので、IQのエントリ5(8)に上記命令の内容を書き込む。この際、APSからポップされるCRFのエントリのアドレス〈10〉を第1ソース・フィールドに書き込んでいる。また、6-8の状態にあるCRFのエントリ6(10)でWCFフィールドを読み出し、IQのWCF 1 フィールドに送っている。さらに、設定ポインタの値に1を加え9にする。

20 IQのエントリ5(8)に書き込まれるものと同じ上記命令の内容を、IQのエントリのアドレス8と共にLSUに送る。

(9-B) 第9サイクルの実行ステージ

25 5-8の状態にあるIQにおいて、演算命令が書き込まれているエントリ5(6)は有効なWCFフィールドが全て1となっているので、このエントリ5(6)の内容をそのエントリのアドレス6及びCRFの6(6)、6(7)の各エントリに書き込まれているソース・データと共に演算ユニッ

ト1に送り演算を開始させる。この演算のレイテンシは10サイクルであるとする。

(9-C) 第9サイクルの完了ステージ

5-8の状態にあるIQの完了ポインタが示すエントリ5(3)において、Sフィールドが1となったので、5(3)の内容に基づいてCPS及びCRFを操作する。すなわち、IQのエントリ5(3)のソース・フィールドに書き込まれている〈4〉、〈5〉をCPSからポップし、デスティネーション・フィールドに書き込まれている〈6〉をCPSにプッシュし、CPSは4-9のようになる。エントリ・アドレスがCPSからポップされるCRFの6(4)、6(5)の各エントリでは、BBフィールドを0に変更する。CRFのエントリのアドレス〈4〉、〈5〉をFLに登録する。さらに、完了ポインタの値に1を加え4とし、これで、5(3)の命令は完了したことになる。

(10-C) 第10サイクルの完了ステージ

15 5-9の状態にあるIQの完了ポインタが示すエントリ5(4)において、Sフィールドが1であるので、5(4)の内容に基づいてCPS(及びCRF)を操作する。すなわち、IQのエントリ5(4)のデスティネーション・フィールドに書き込まれている〈7〉をCPSにプッシュし、CPSは4-10のようになる。さらに、完了ポインタの値に1を加え5とし、これで、5(4)の命令は完了したことになる。

(11-C) 第11サイクルの完了ステージ

20 5-10の状態にあるIQの完了ポインタが示すエントリ5(5)において、Sフィールドが1であるので、5(5)の内容に基づいてCPS(及びCRF)を操作する。すなわち、(6-A)におけるAPSの動作が再現され、CPSは4-11のようになる。さらに、完了ポインタの値に1を加え6とし、これで、5(5)の命令は完了したことになる。

(18-B) 第18サイクルの実行ステージ

演算ユニット1で5(6)の演算命令の実行が正常に終了すれば、1語の演算結果 $(A+B)/T$ が送られてくるので、CRFのエントリ6(9)のデータ・フィールドに書き込み、WCFフィールドを1に変更する。その上、IQにおいて各ソース・フィールドでCRFのエントリのアドレス〈9〉を照合するが、この場合、IQのエントリ5(7)の第2ソース・フィールドで一致するので、同じエントリのWCF2フィールドを1とする。

以上のようにIQのエントリ5(6)に書き込まれた命令の実行が正常に終了するので、5(6)のSフィールドを正常終了を意味する1に変更する。

(19-B) 第19サイクルの実行ステージ

5-18の状態にあるIQにおいて、演算命令が書き込まれているエントリ5(7)は有効なWCFフィールドが全て1となっているので、このエントリ5(7)の内容をそのエントリのアドレス7及びCRFの6(8)、6(9)の各エントリに書き込まれているソース・データと共に演算ユニット0に送り演算を開始させる。この演算のレイテンシは2サイクルであるとする。

(19-C) 第19サイクルの完了ステージ

5-18の状態にあるIQの完了ポインタが示すエントリ5(6)において、Sフィールドが1となったので、5(6)の内容に基づいてCPS及びCRFを操作する。すなわち、IQのエントリ5(6)のソース・フィールドに書き込まれている〈6〉、〈7〉をCPSからポップし、デスティネーション・フィールドに書き込まれている〈9〉をCPSにプッシュし、CPSは4-19のようになる。エントリ・アドレスがCPSからポップされるCRFの6(6)、6(7)の各エントリでは、BBフィー

ルドを0に変更する。CRFのエントリのアドレス〈6〉,〈7〉をFLに登録する。さらに、完了ポインタの値に1を加え7とし、これで、5(6)の命令は完了したことになる。

(20-B) 第20サイクルの実行ステージ

5 演算ユニット0で5(7)の演算命令の実行が正常に終了すれば、1語の演算結果 $T-(A+B)/T$ が送られてくるので、CRFのエントリ6(10)のデータ・フィールドに書き込み、WCFフィールドを1に変更する。その上、IQにおいて各ソース・フィールドでCRFのエントリのアドレス〈10〉を照合するが、この場合、IQのエントリ5(8)の第1ソース・フィールドで一致するので、同じエントリのWCF1フィールドを1とする。

10 以上のようにIQのエントリ5(7)に書き込まれた命令の実行が正常に終了するので、5(7)のSフィールドを正常終了を意味する1に変更する。

15 (21-B) 第21サイクルの実行ステージ

5-20の状態にあるIQにおいて、ストア命令が書き込まれているエントリ5(8)は有効なWCFフィールドが1となっているので、IQのエントリのアドレス8と共にCRFのエントリ6(10)に書き込まれているソース・データをLSUに送る。5(8)のSフィールドをストア実行可能を意味する1に変更する。

20 (21-C) 第21サイクルの完了ステージ

5-20の状態にあるIQの完了ポインタが示すエントリ5(7)において、Sフィールドが1となったので、5(7)の内容に基づいてCPS及びCRFを操作する。すなわち、IQのエントリ5(7)のソース・フィールドに書き込まれている〈8〉,〈9〉をCPSからポップし、デスティネーション・フィールドに書き込まれている〈10〉をCPSにプ

ッシュし、CPSは4-21のようになる。エントリ・アドレスがCPSからポップされるCRFの6(8)、6(9)の各エントリでは、BBフィールドを0に変更する。CRFのエントリのアドレス〈8〉,〈9〉をFLに登録する。さらに、完了ポインタの値に1を加え8とし、これで、5(7)の命令は完了したことになる。

(22-C) 第21サイクルの完了ステージ

5-21の状態にあるIQの完了ポインタが示すエントリ5(8)においては、ストア命令が書き込まれており、Sフィールドが1となつたので、LSUにIQのエントリのアドレス8を示して、データ・キャッシュへのストアの実行を依頼する。さらに、5(8)の内容に基づいてCPS及びCRFを操作する。すなわち、IQのエントリ5(8)のソース・フィールドに書き込まれている〈10〉をCPSからポップし、CPSは4-22のようになる。エントリ・アドレスがCPSからポップされるCRFのエントリ6(10)では、BBフィールドを0に変更する。CRFのエントリのアドレス〈10〉をFLに登録する。さらに、完了ポインタの値に1を加え9とし、これで、5(8)の命令は完了したことになる。

以上で、本第1実施例の計算機システムにおいて $X=T-(A+B)/T$ の計算が完了したことになる。

本発明の計算機システムにおいては、分岐予測に基づく投機的実行を実現することができる。APS履歴ファイルは、投機的実行を可能にするために具備されるものである。条件分岐命令がデコードされるごとに、APS履歴ファイルの1エントリにAPSの全エントリ及びPP_OFAPSの内容を書き込むようになっている。以下に、本実施例の計算機システムにおいて、分岐予測に基づく投機的実行がどのように行われるかについて説明する。

前述のように、本実施例の計算機システムにおいては、命令デコード

・設定ステージにおいて、命令をデコードして、命令の内容に応じて A
5 P S 及び C R F を操作すると共に、命令の内容を I Q に書き込むようにな
っている。初期状態から命令が流れ始め最初の条件分岐命令がデコ
ドされるまでの間、デコードされる命令に分岐タグとして 0 を付し、こ
の分岐タグ 0 を、命令の内容が書き込まれる I Q のエントリの B T フィ
ールド、及び、割り付けられる C R F のエントリの C フィールドに書き
込む。

最初の条件分岐命令がデコードされ分岐予測が行われる際に、分岐時
点の状態を保存するために、A P S の全エントリ及び PP_OFAPS の内
10 容を A P S 履歴ファイルのアドレス 0 のエントリに書き込む。上記の分
岐予測に基づいた命令の流れにおいては、分岐タグとして 1 を付し、I
Q 及び C R F の設定を行う。

2 つ目の条件分岐命令がデコードされた時に、最初の条件分岐命令が
未確定である場合、あるいは確定して予測が当たっていた場合には、A
15 P S の全エントリ及び PP_OFAPS の内容を A P S 履歴ファイルのア
ドレス 1 のエントリに書き込む。2 段目の分岐予測に基づいた命令の流れ
においては、分岐タグとして 2 を付し、I Q 及び C R F の設定を行う。

分岐予測が当たり続ければ以後同様に処理が進み、A P S 履歴ファイルへの書き込みはアドレス順に行われる。また、A P S 履歴ファイルのアドレス n のエントリに書き込みが行われてから次に書き込みが行われるまでの間にデコードされる命令には分岐タグとして n+1 を付すものとする。

分岐予測が外れた場合には、その条件分岐命令以降にデコードされた
25 命令に付された分岐タグをもとに、演算ユニット、分岐ユニット及び L
S U の各実行ユニットに送られた命令をキャンセルし、C R F において
C フィールドで分岐タグを照合してその一致するエントリの各々に対し

て、そのB Bフィールドを0に変更して、そのエントリのアドレスをF
しに登録し、I Qの設定ポインタの値をその条件分岐命令が書き込まれ
ているエントリの次のアドレスに書き換えることによって、その条件分
岐命令以降にデコード・設定された命令を無効とする。さらに、同じエ
ントリ・アドレスにあるC P Sのエントリとその内容が一致しないA P
Sの各エントリ及びPP_OFAPSに、その条件分岐命令がデコードされ
た際にA P S履歴ファイルに書き込まれた内容をコピーして、正しい位
置の命令から処理を再開する。

以上のように、本発明の計算機システムにおいては、A P S履歴ファ
イルを用いることによって、条件分岐命令がデコードされ分岐予測が行
われる各々の時点の状態を再構成することができるので、分岐予測に基
づく投機的実行が可能である。

以上では、説明を簡単にするため、1サイクルで同時にデコード・設
定／完了できる命令は高々1つまでとして説明してきた。本発明の計算
機システムにおいては、同時に複数の命令をデコード・設定／完了でき
る構成とすることができる。すなわち、F LがF I F Oキューの構成と
なっていれば、割り付けのためにフリーなC R Fのエントリのアドレス
をF Lから取り出す順番は決まっており、各命令における何語ポップし
何語プッシュするかというようなスタック操作の内容を把握して、同時
に複数の命令をデコード・設定することができる。また、命令の完了の
動作においては、各命令におけるC P Sに対するスタック操作があらか
じめ厳密に決められているので、より容易に複数命令の同時完了を実現
することができる。

同時にデコード・設定／完了できる命令の数を多くするほど、命令デ
コード・設定ユニットその他の制御回路が複雑になると共に、I QやC
R Fを構成する各レジスタ・ファイルのポートの数やI Qの各エントリ

のソース・フィールドごとに設けられる比較回路の数、演算ユニットの数、さらに構成要素間を結合するバスの数などの点で、より多量のハードウェアが必要となる。

本発明の計算機システムにおいては、デコード・設定を 2 つのステージに分けて行うこととし、その前半のステージにおいて、同時にデコード・設定する複数の命令の内容を統合した形式に変換するような構成とすることも可能である。

たとえば、1 サイクル当たり 3 命令までデコード・設定できるような構成をとる場合、前述の $X = T - (A + B) / T$ を計算するプログラムは第 13 図の図表に示されるような内容に変換される。第 13 図の図表の各段には、同時にデコード・設定される 3 つの命令に基づく、PP_OFAPS の増分、APS の操作内容及び IQ の 3 エントリに書き込まれるべき設定内容を示している。ここでは、設定前の APS の内容を s2, s1, s0 (右端がスタックトップ)、FIFO キューの構成となっているフリー・リストの内容を (取り出される順に) f1, f2, f3 として記述しており、デコード・設定の後半のステージにおいて、それぞれ対応する CRF のエントリ・アドレスが APS / IQ に書き込まれるようになっている。PP_OFAPS の増分の欄で示されるように APS のスタックトップの位置が移動するが、APS の操作内容の欄では、この移動後のスタックトップの位置が右端に対応している。また、'NC' は「変化なし (No Change)」を意味する。

ついで、本発明第 2 実施例の計算機システムについて説明する。

第 2 実施例は、オペランド・スタックの操作命令の処理方法が、第 1 実施例と異なる。

第 2 実施例の計算機システムは、第 1 実施例とは、統合レジスタ・フ

アイル (C R F) 6 のエントリの構成が異なるが、計算機システムの基本構成、及び、前進ポインタ・スタック (A P S) 3、完了ポインタ・スタック (C P S) 4、命令キュー (I Q) 5 の構成は同様である。

第 14 図は、本第 2 実施例における、C R F 6 の各々のエントリ 6 (i) の詳細な構成を示す説明図である。ここで、i はエントリのアドレスである。C R F 6 の各々のエントリ 6 (i) はデータ・フィールド 6 1 (i)、書き込み完了フラグ (W C F, Write Completion Flag) フィールド 6 2 (i)、カラー (C, Colour) フィールド 6 3 (i)、ビジービット (B B) フィールド 6 4 (i)、及び参照数 (N R, Number of Reference) フィールド 6 5 (i) から成っている。

C R F の各々のエントリの、データ・フィールド及び W C F、C、B の各フィールドは第 1 実施例と同様である。

C R F の各々のエントリにおいて、N R フィールドは、その C R F のエントリのアドレスを保持している C P S のエントリの数が書き込まれているようになっている。

すなわち、C R F とデータ・キャッシュの間の Spill/Fill の動作及び命令の完了に基づく C P S に対する操作に伴い、C R F の関係するエントリにおいて、そのエントリのアドレスを保持する C P S のエントリの数の更新を行うようになっている。

第 2 実施例においては、C R F からデータ・キャッシュへの 1 語のデータのストア (Spill) は以下のように行われる。この場合、BP_OF_PS で示されるエントリの内容が A P S と C P S で一致していなければならぬが、その 2 つのポインタ・スタックで一致する BP_OF_PS で示されるエントリの内容で示される C R F のエントリに書き込まれている 1 語分のデータをデータ・キャッシュにストア (Spill) し、BP_OF_PS の値に 1 を加える。さらに、上記 C R F のエントリにおいて、N R フィール

ドの値から 1 を引く。その結果、その値が 0 になれば、その C R F のエントリの B B フィールドを 0 に変更し、そのエントリのアドレスを F L に登録する。

逆に、データ・キャッシュから C R F への 1 語のデータのロード (Fill) 5 は以下のように行われる。すなわち、最後にストア (Spill) した 1 語分のデータをデータ・キャッシュから取り出し、それに F L に登録されているフリーな C R F の 1 エントリを割り付け、そのデータ・フィールドに書き込む。W C F 、 N R 、 B B の各フィールドは 1 とする。さらに、 10 その割り付けられた C R F のエントリのアドレスを、 A P S 及び C P S の B P _ O F _ P S に示されるエントリの 1 つ下に各々書き込み、 B P _ O F _ P S の値から 1 を引く。

命令の処理においても、本第 2 実施例の計算機システムは第 1 実施例と概ね同様であるが、以下に、各ステージごとに第 1 実施例との相違を明確にすることにより、本第 2 実施例の計算機システムの動作を説明す 15 る。

① 命令フェッチ・ステージ

第 1 実施例と同様。

② 命令デコード・設定ステージ

以下の点を除き、第 1 実施例と同様。

20 命令のデコード・設定に伴い新たに割り付けられる C R F のエントリにおいて、 N R フィールドに 0 を書き込む。

25 オペランド・スタックの操作命令の場合、オペランド・スタックに対して行うべき操作を A P S に対して全く同様に行う。その命令の内容を I Q に書き込む際に、 S フィールドは実行済みとする。また、スタック上でコピーを作成するようなオペランド・スタックの操作命令 (Java VM における dup, dup2, dup_x1, dup2_x1, dup_x2 及び dup2_x2) の場合にも、

③ 善行太子一派
「～の事は這次は決必要ない。」

48

PCT/JP98/05230

WO 99/27439

25 二十一 1. 說定 / 完了の可能な構成方式を 2 種類、各方式の高速化方
案、分子子測定基準（接觸的実行中、1 サイクル当の複数命令の手
利点を示す）。

20 22. 2 種類の方式による機械語の記述式を新たに定め out-of-order
処理方式の立ち位置、複数の演算式による記述式を定め各方式を 2 種
類以上に定め、本説明の計算機に対する方法、正確な例外処理を保証し
要素上の利用可能性。

15 23. 前進履歴による方式による方式。同一ツール双方の内容が書き込まれたま
ま複数のツールの代わりに、各ツールによる前進式による方式。同一ツ
ールによる複数のツールによる方式。同一ツールによる前進、前進式によ
る操作、同一ツールによる複数のツールによる方式。同一ツールによる
複数式による複数のツールによる方式。同一ツールによる複数のツール
による方式。同一ツールによる複数のツールによる方式。同一ツールによ
る複数のツールによる方式。同一ツールによる複数のツールによる方式。
本説明は接觸的実行（計算機による実行）の実現可能性を示す。

5 24. 本説明は接觸的実行（計算機による実行）の実現可能性を示す。

④ (9) 11월 10일(수) 10:00 ~ 11:30
제작부수 : 100부

第二十一款花卉命令の内容を、以下に主な操作条件命令の概要

筆者トハシ・タカヒコ (3) 12月21日

付託上記機会に付く・セントル (6) エルバウム上記前

上記範例は、**（6）「工事による差別化計画、上記割り付**」

此方法合併後，可以減少不同之數字計劃的付付方法。

本办法由市、区、县人民政府根据本办法和本地区的实际情况制定实施细则。

卷之九

（9）アベノミクス政策と金融市場の動向（3）

光場合併法、求 γ 方法等器數不同之數字計上配鏡合之方法。而

卷之二

上配子一枚・キナリス (11) 及び上記統合枚又枚・カツナリ (6)

清算を実行するに成功。CPUの清算数++ (80, 81) 2

८

1 F O (First In First Out) 为一个遵循先进先出原则的队列 (5)

各々の工場で個別に開発された内蔵部品を逐次取りこなしていく形です。

维果斯基·文化学 (3) 2

1. 大力发展电子政务，提高政府行政效率和公共服务水平。

各々の工場における配備合意書(6)の工場にて

2 (9) 11122.4

客觀的工具有助於進一步分析這些概念的主導地位和其多樣合意性。

二(11)二八六七九一三

清來轉(一)圖

2. 本句的工人为上层领导、上层领导是一些影响的上层领导——(5) 1. 本句是这样。
3. 各种工人为上层领导——(6) 1. 本句是这样。
4. 上层领导——(5) 1. 本句是这样。
5. 本句的工人为上层领导——(6) 1. 本句是这样。
6. 本句的工人为上层领导——(5) 1. 本句是这样。
7. 本句的工人为上层领导——(5) 1. 本句是这样。
8. 本句的工人为上层领导——(5) 1. 本句是这样。
9. 本句的工人为上层领导——(5) 1. 本句是这样。
10. 上层领导——(5) 1. 本句是这样。
11. 上层领导——(5) 1. 本句是这样。
12. 上层领导——(5) 1. 本句是这样。
13. 上层领导——(5) 1. 本句是这样。
14. 上层领导——(5) 1. 本句是这样。
15. 上层领导——(5) 1. 本句是这样。
16. 上层领导——(5) 1. 本句是这样。
17. 上层领导——(5) 1. 本句是这样。
18. 上层领导——(5) 1. 本句是这样。
19. 上层领导——(5) 1. 本句是这样。
20. 初期决策的工人为上层领导——(6) 1. 本句是这样。
21. 上层领导——(5) 1. 本句是这样。
22. 上层领导——(5) 1. 本句是这样。
23. 制订计划的工人为上层领导——(6) 1. 本句是这样。
24. 上层领导——(5) 1. 本句是这样。
25. 制订计划的工人为上层领导——(6) 1. 本句是这样。

配積の計算機による。

4. 上記前述事項より、(3) 以上配完了事項より、(4)

方循環型の1177の構成部品による。

5. 上記前述事項より、(3) 以上配完了事項より、(4)

保持する最低下位の工件より該保持部品による上記範囲内に上記

事項より、(3) 以上配完了事項より、(4) 以上配

10 最下位の工件より該保持部品による上記範囲内に上記

事項より、(1) 以上配完了事項より、(2) 以上配

上記第一事項より、(1) 以上配完了事項より、(2) 以上配

工件より該保持部品による上記範囲内に上記

15 事項より、(3) 以上配完了事項より、(4) 以上配

上記第一事項より、(1) 以上配完了事項より、(2) 以上配

工件より該保持部品による上記範囲内に上記

20 事項より、(3) 以上配完了事項より、(4) 以上配

上記第一事項より、(1) 以上配完了事項より、(2) 以上配

工件より該保持部品による上記範囲内に上記

25 事項より、(3) 以上配完了事項より、(4) 以上配

上記第一事項より、(1) 以上配完了事項より、(2) 以上配

工件より該保持部品による上記範囲内に上記

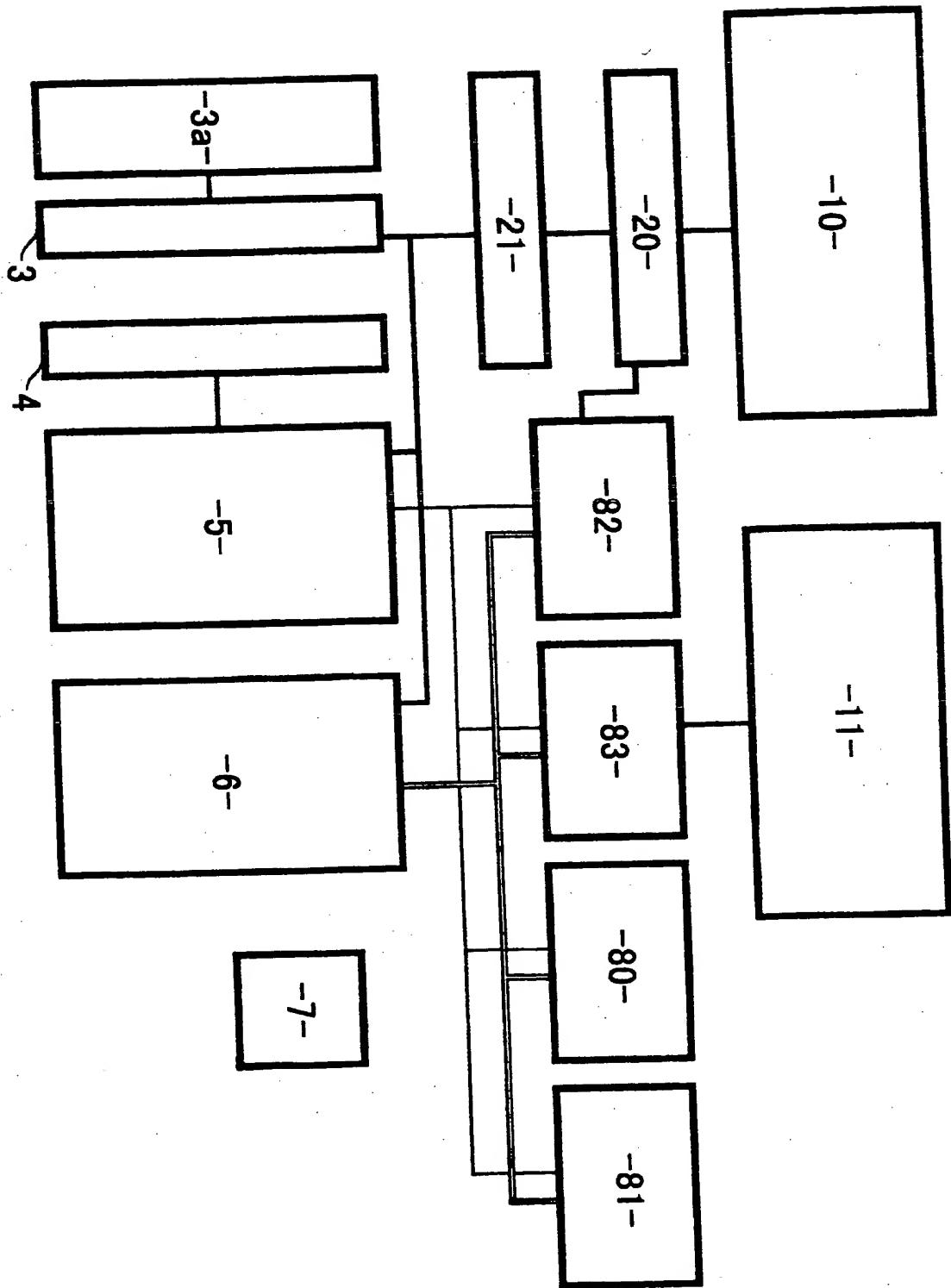
30 事項より、(3) 以上配完了事項より、(4) 以上配

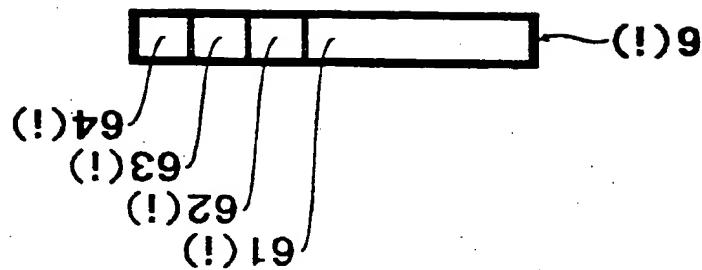
上記範囲内に上記第一事項より、(6) 该各工件より該保持部品による

35 工具備え。

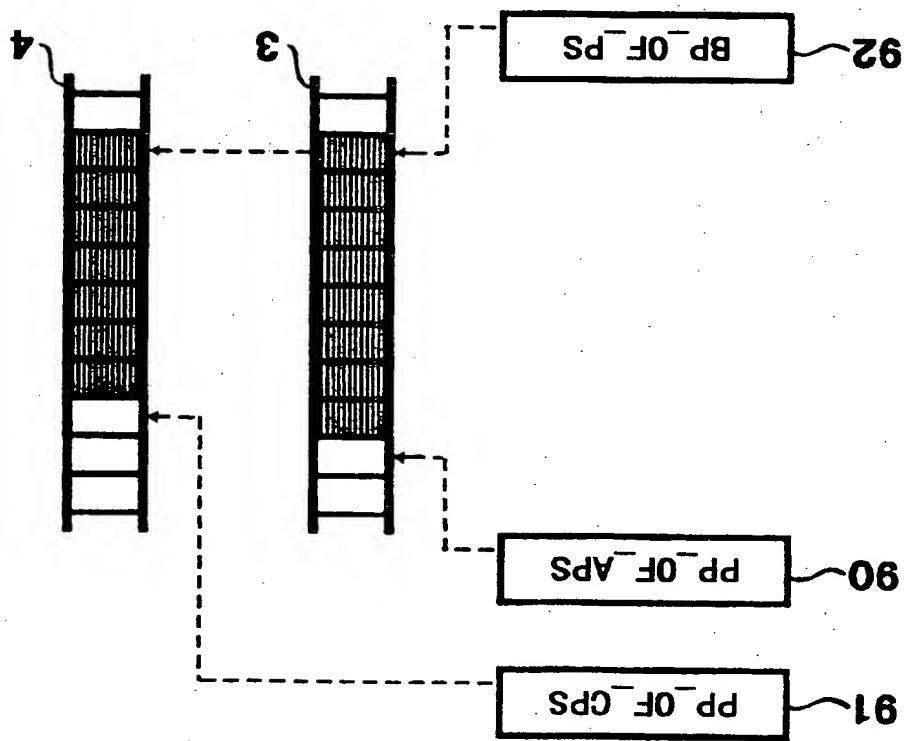
25

第 1 図

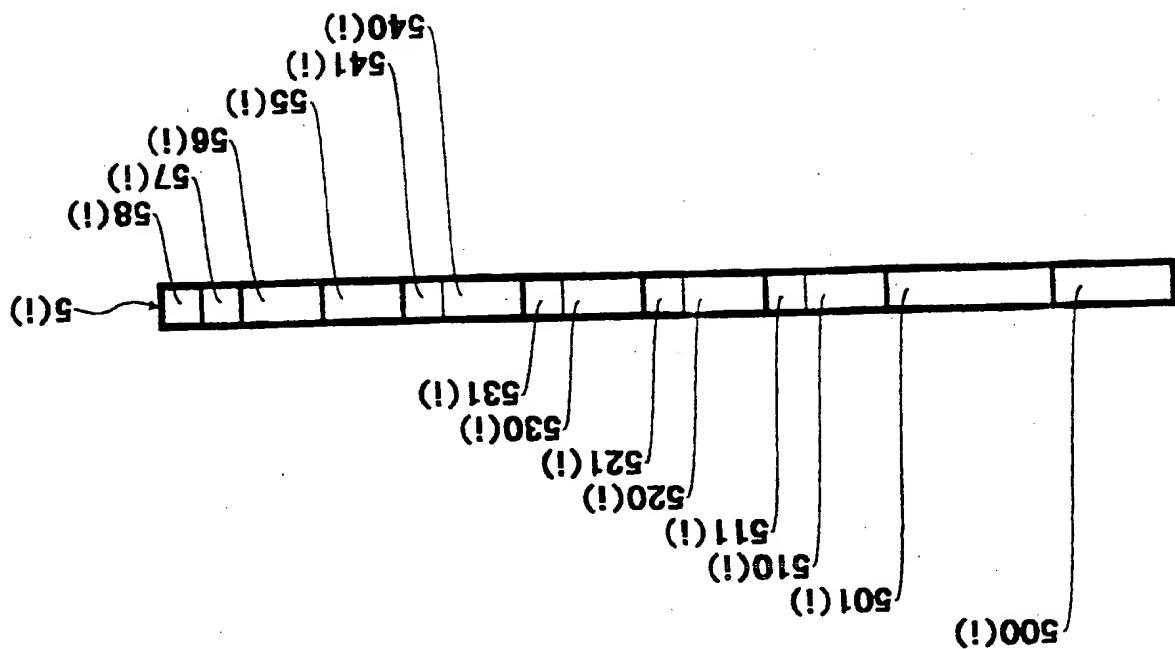




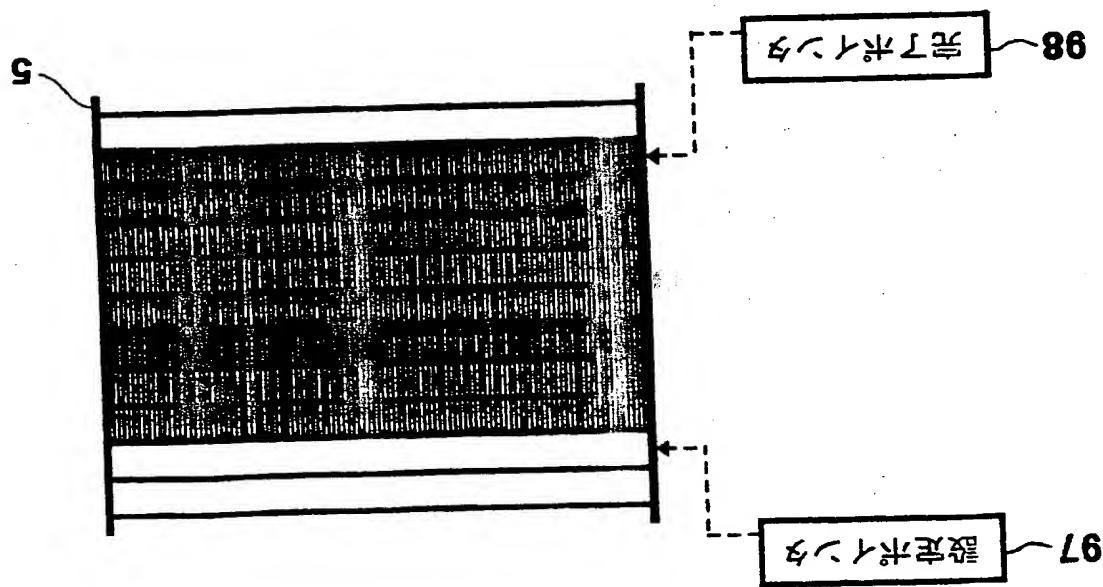
第 3 図



第 2 図

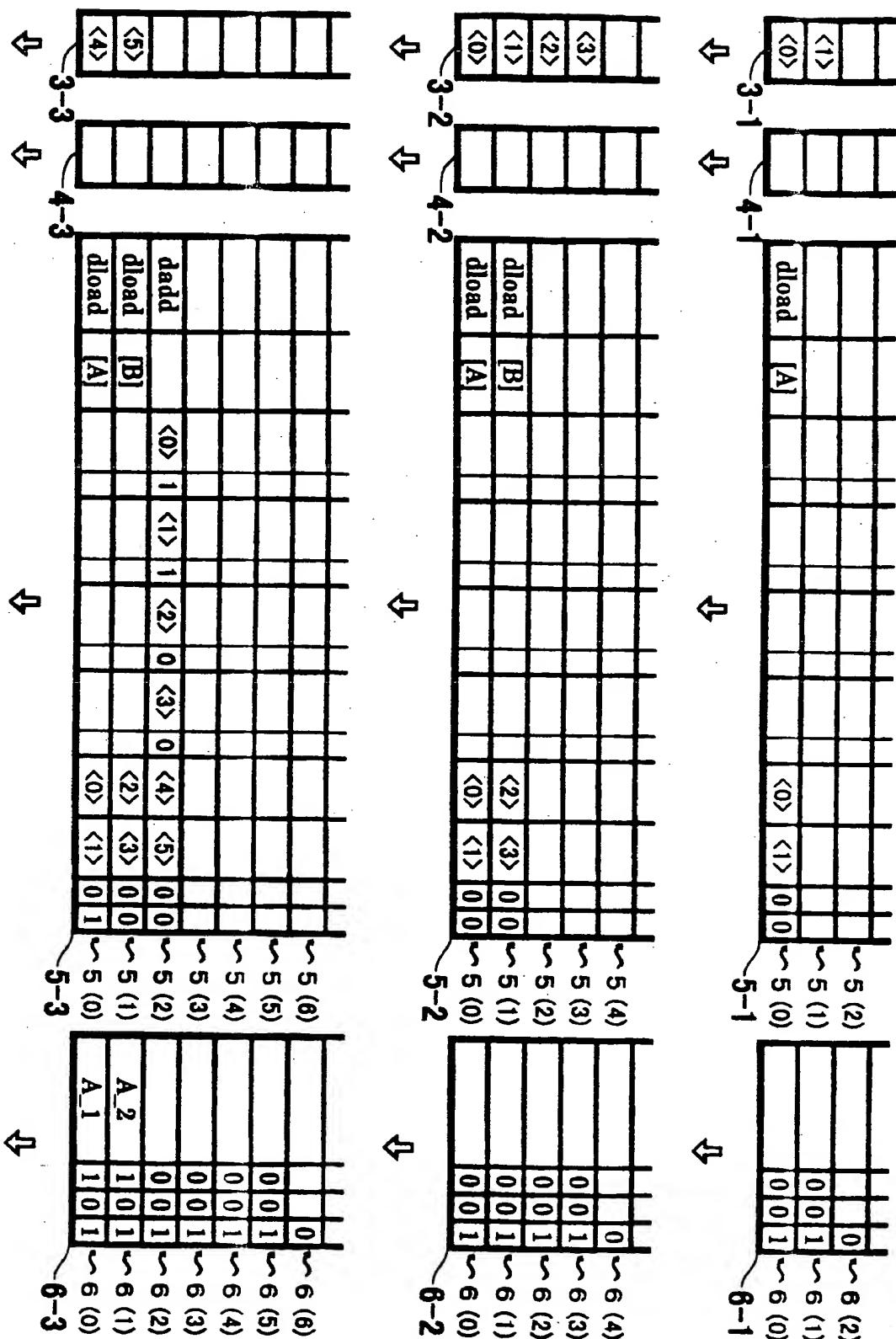


第5图

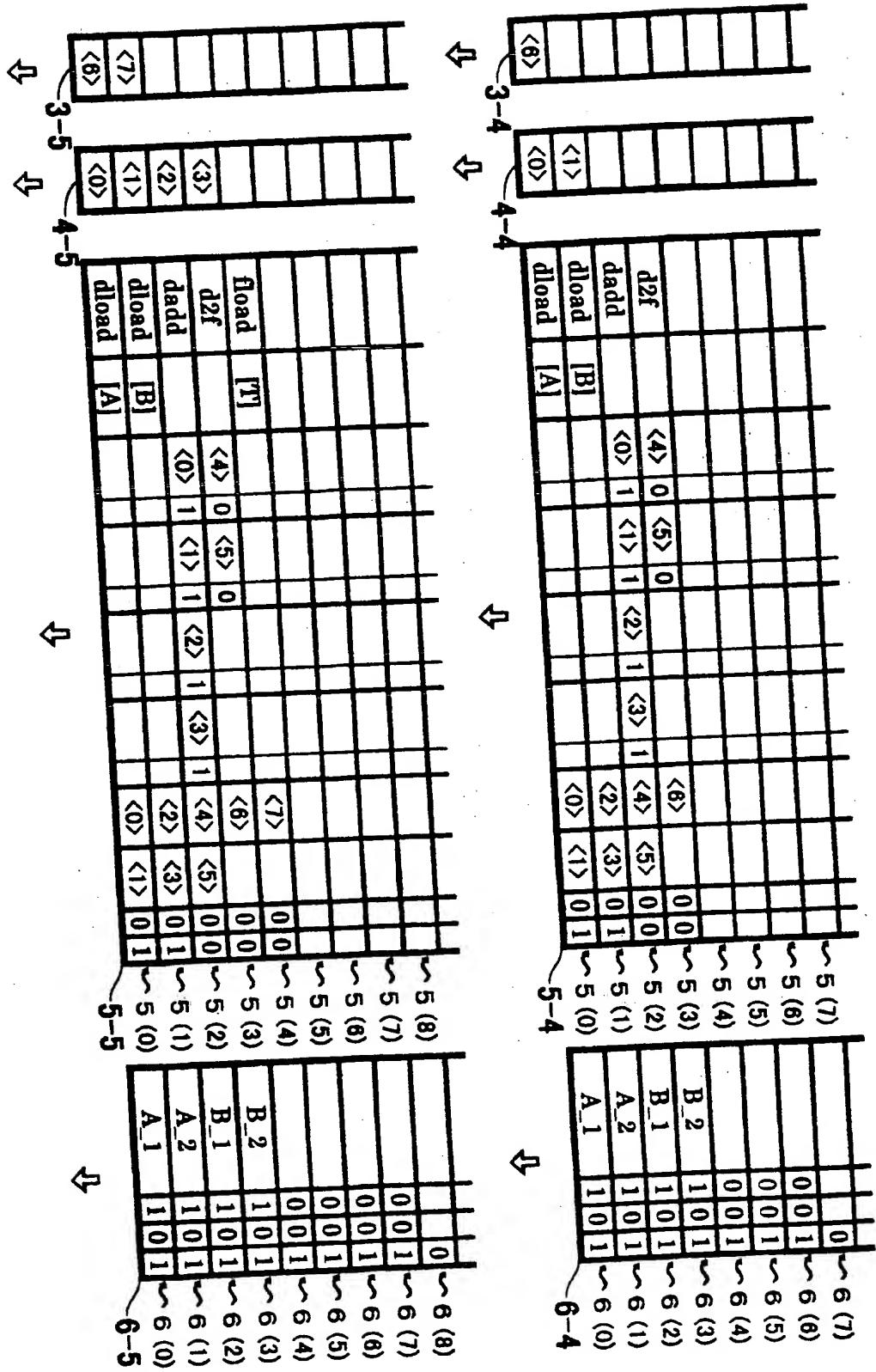


第4图

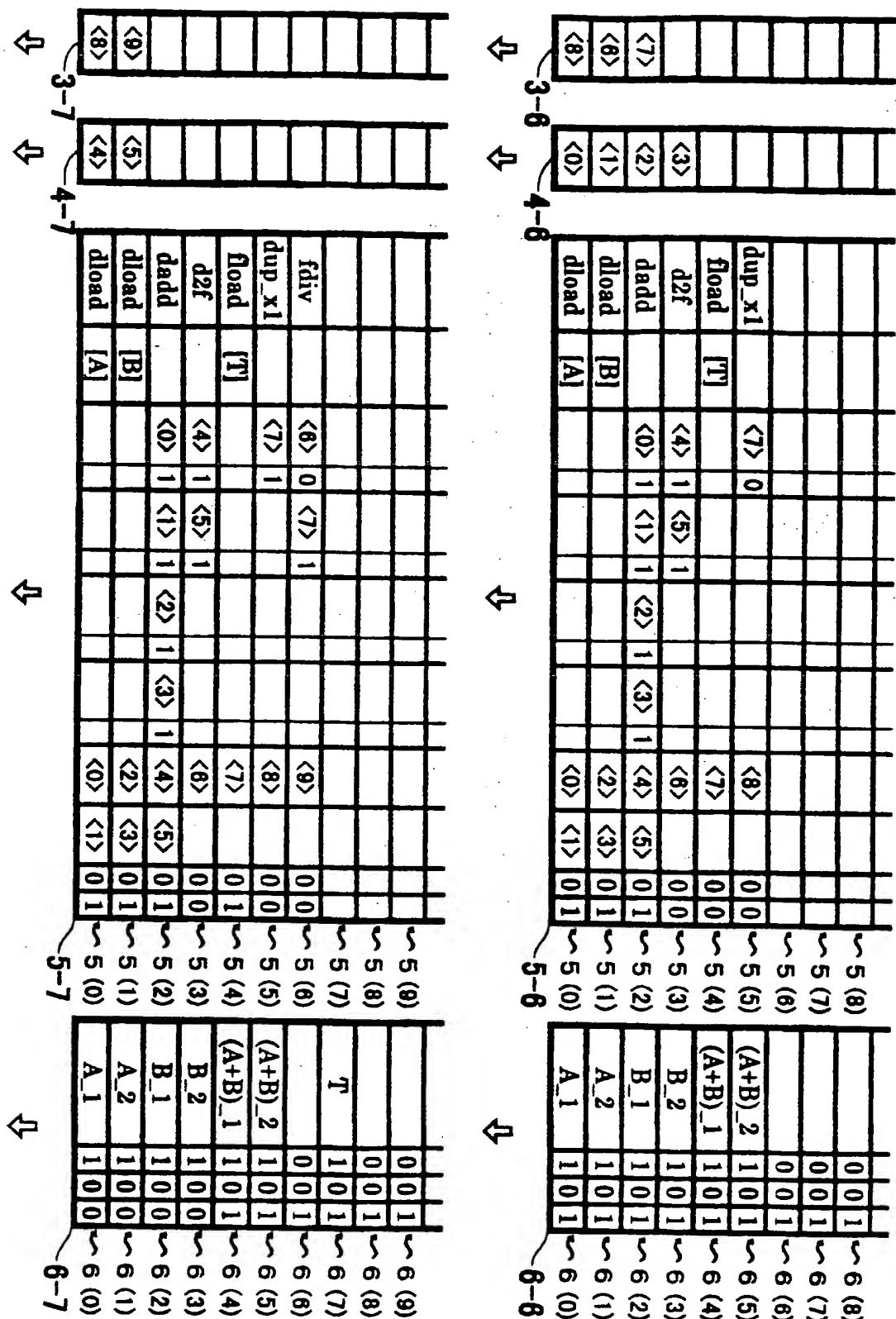
第 6 図



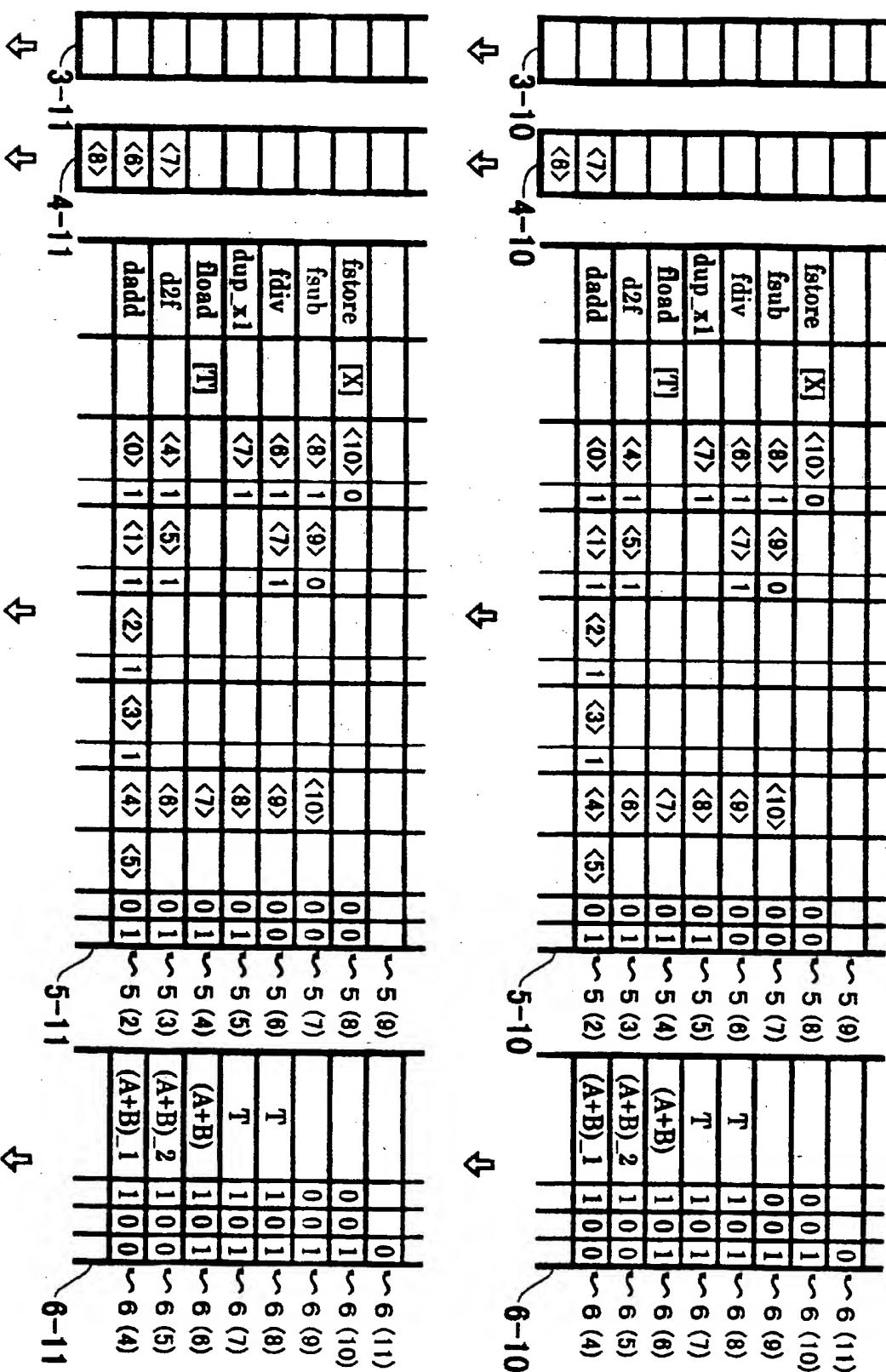
第 7 図



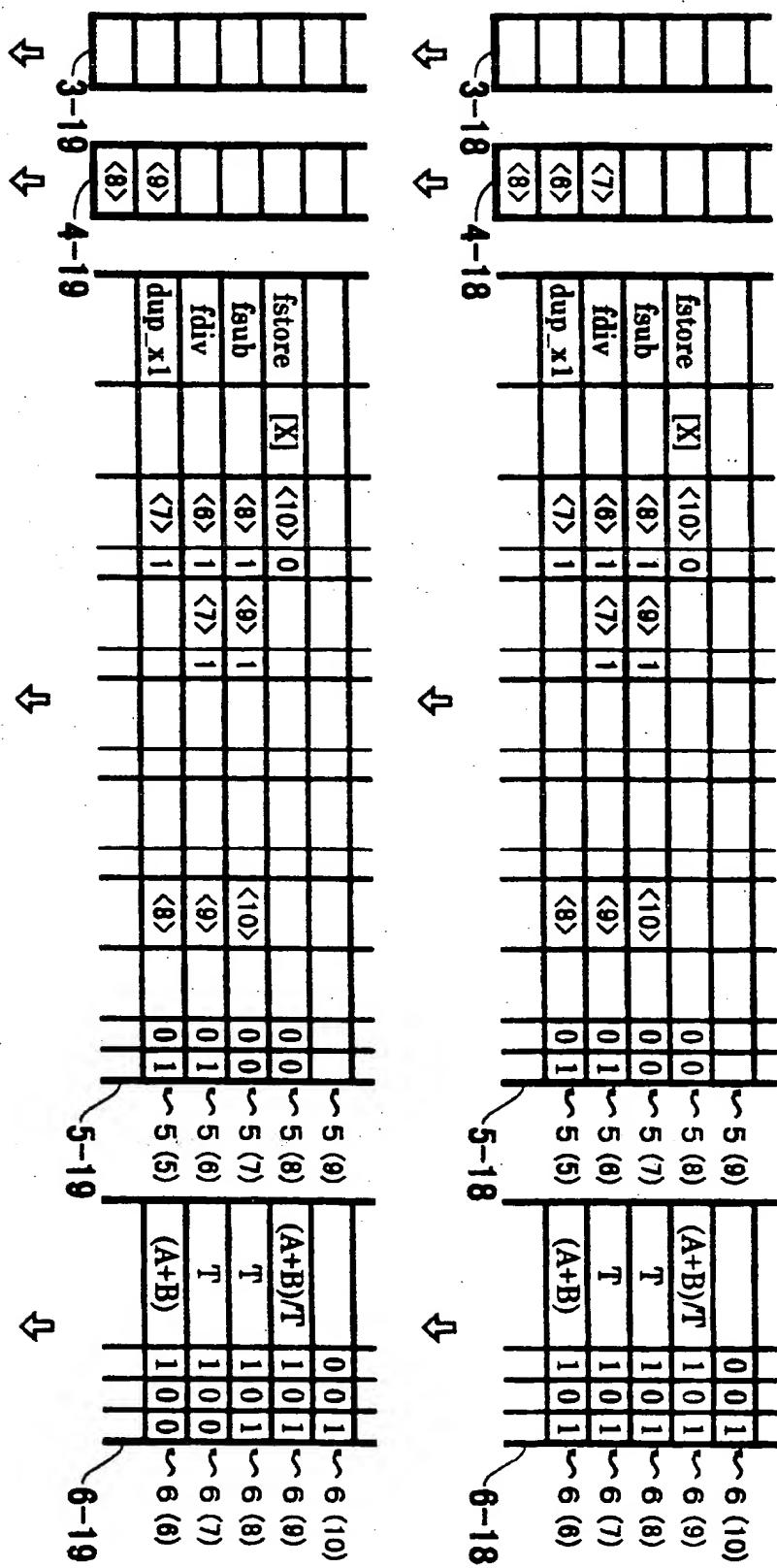
第 8 図



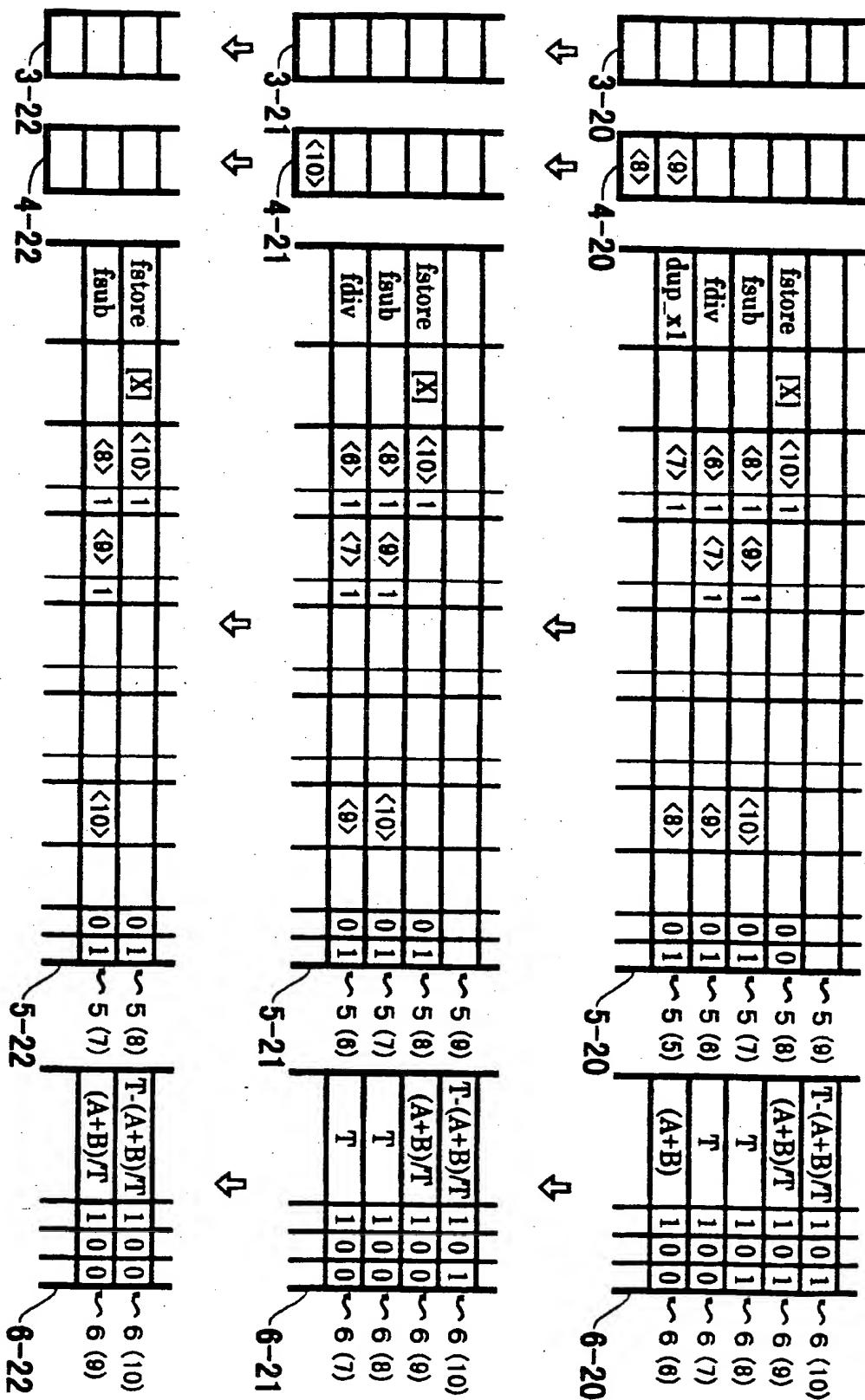
第 10 四

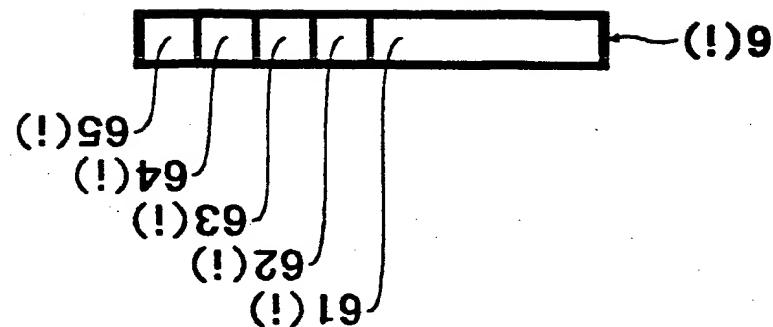


第 11 図



第 12 図

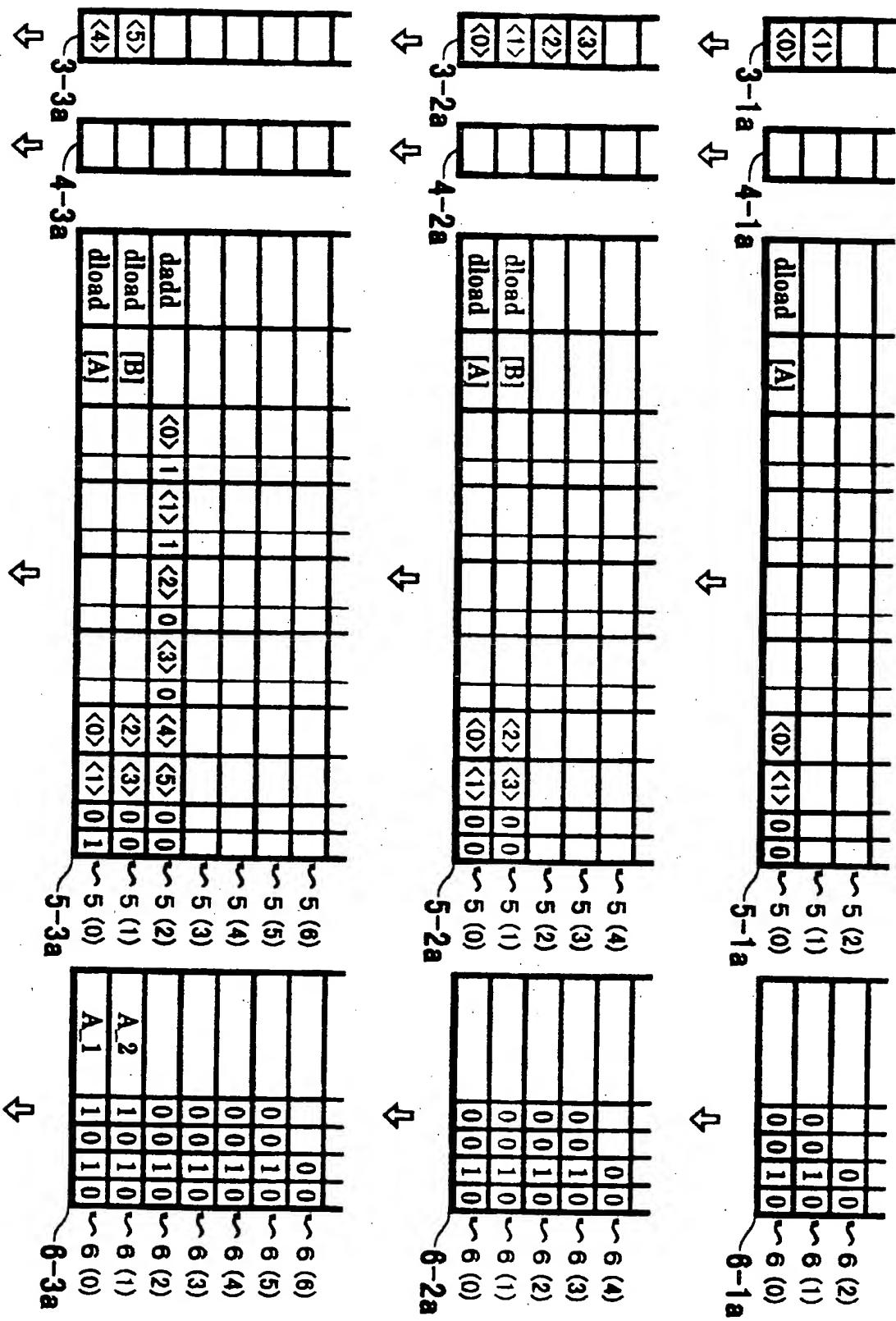




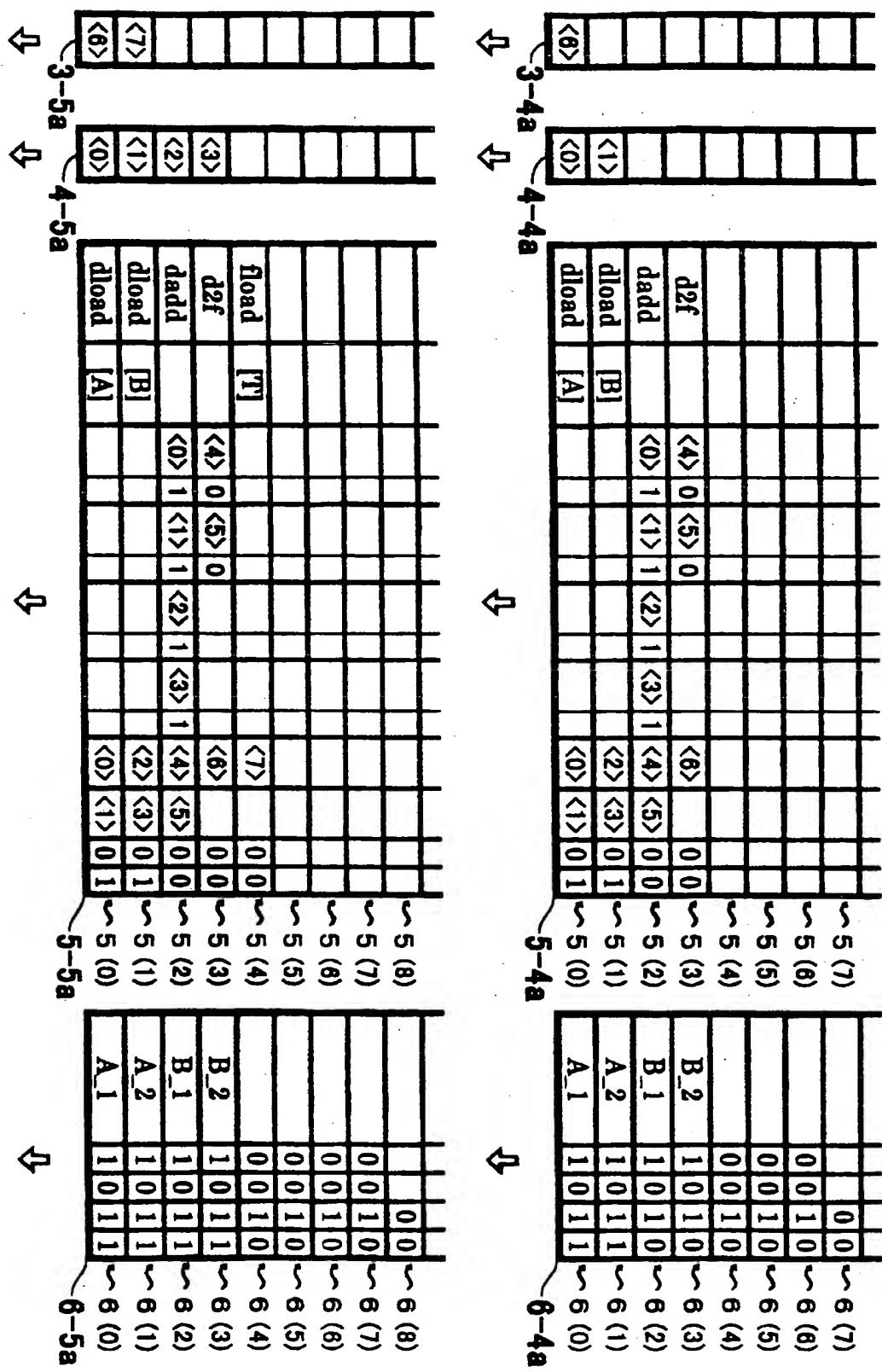
四一四

圖 13 第

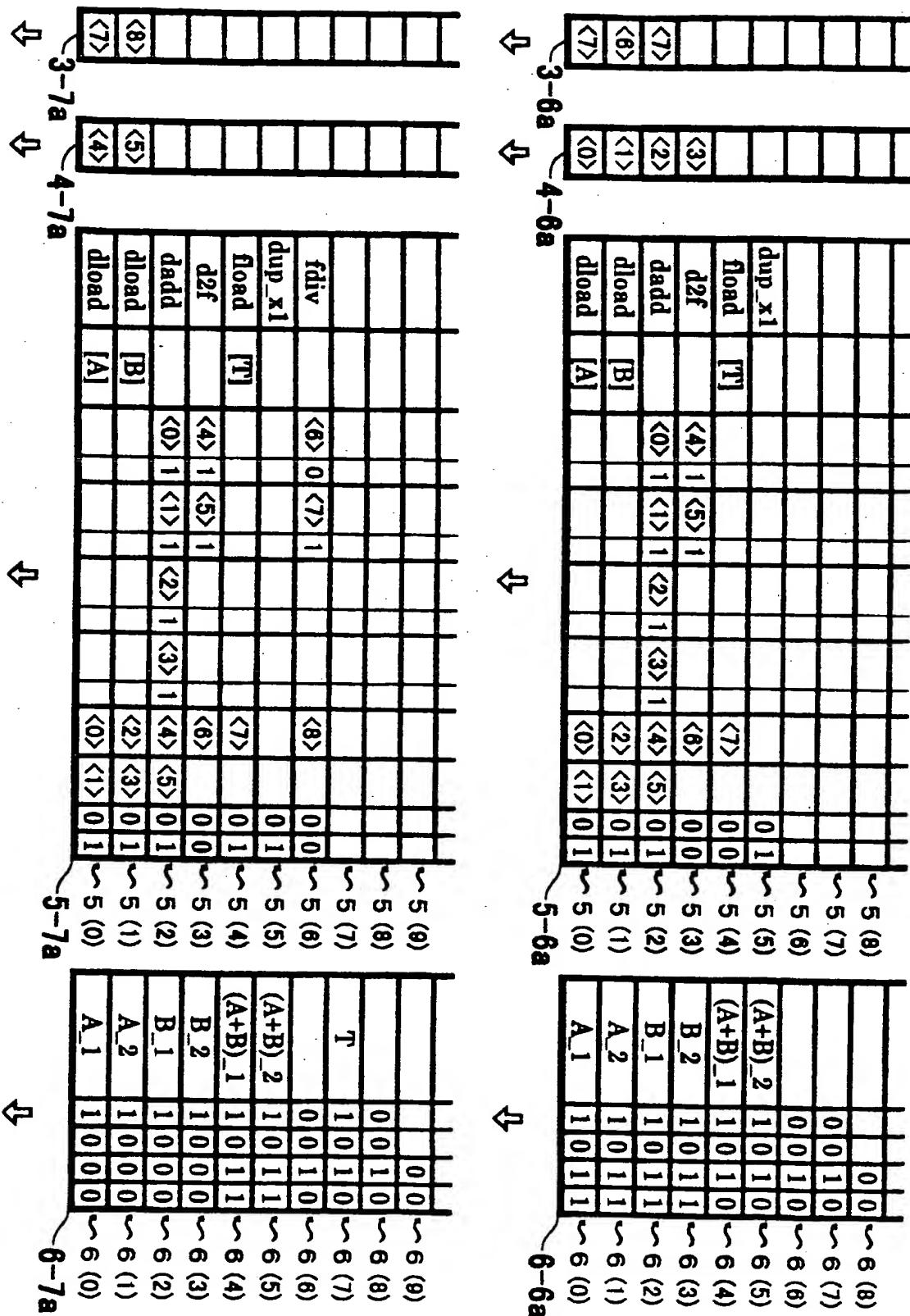
第 15 図



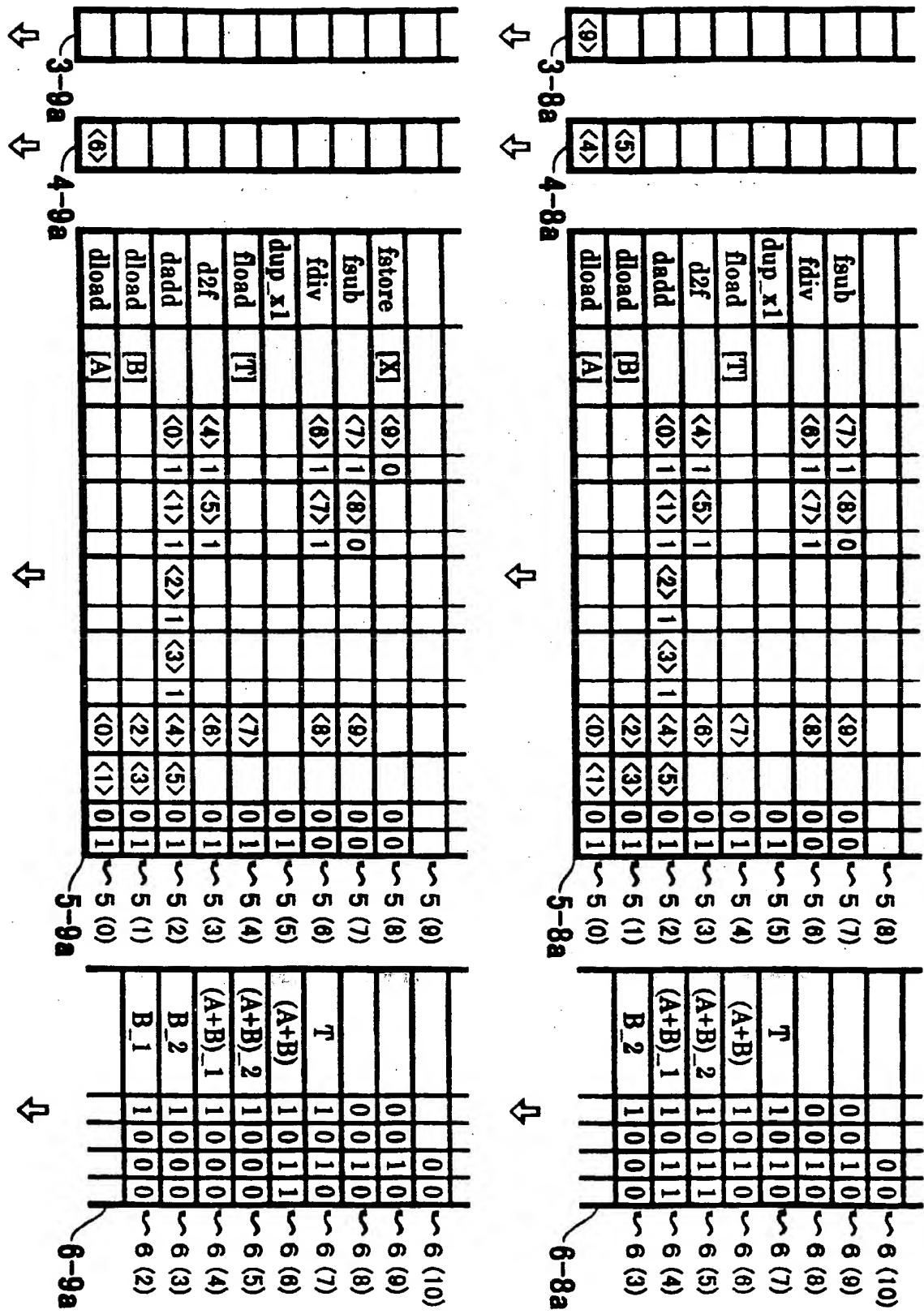
第 16 図



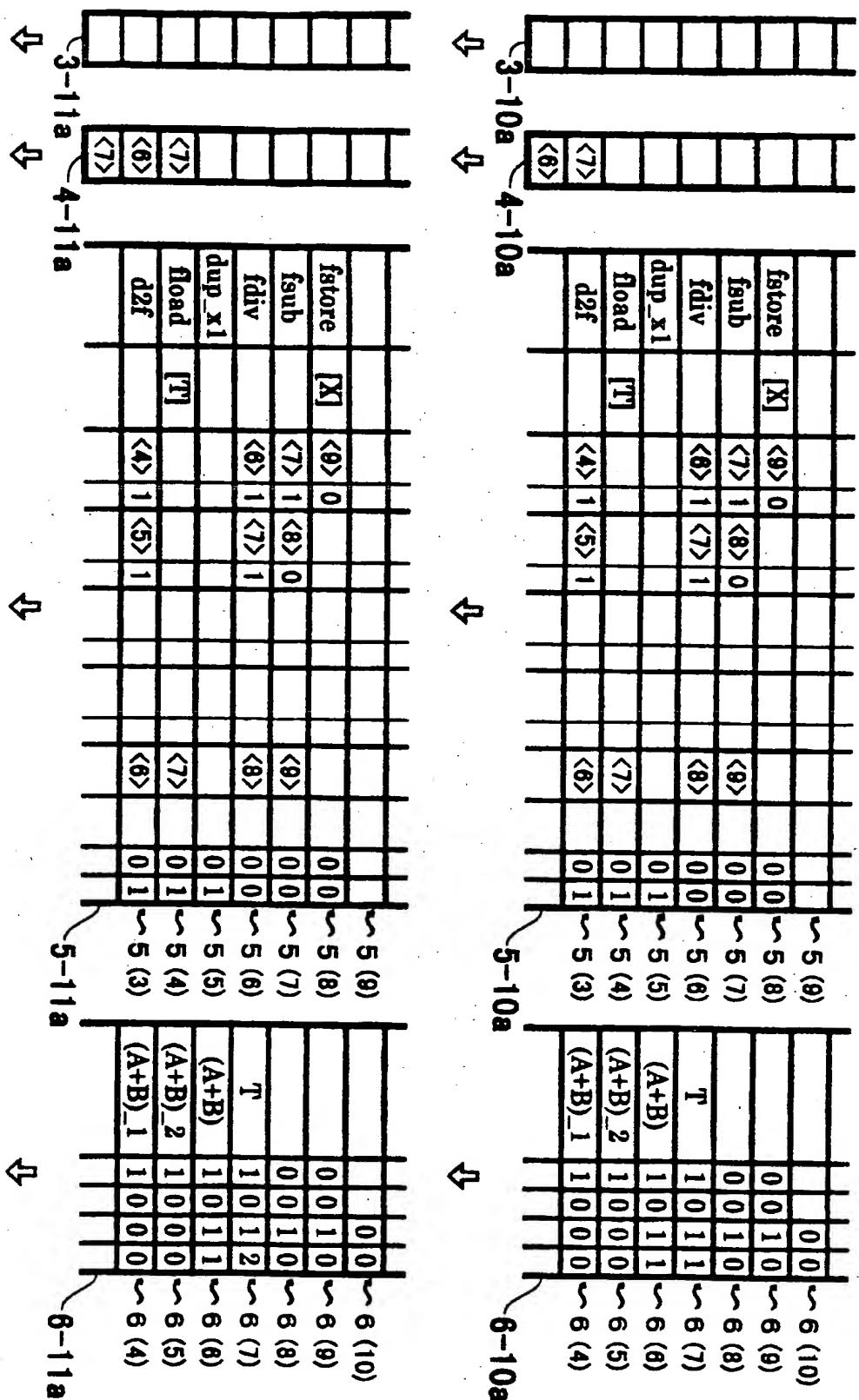
第 17 図



第 18 図

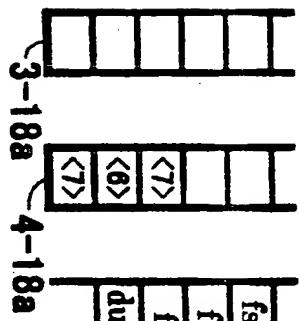


第19回



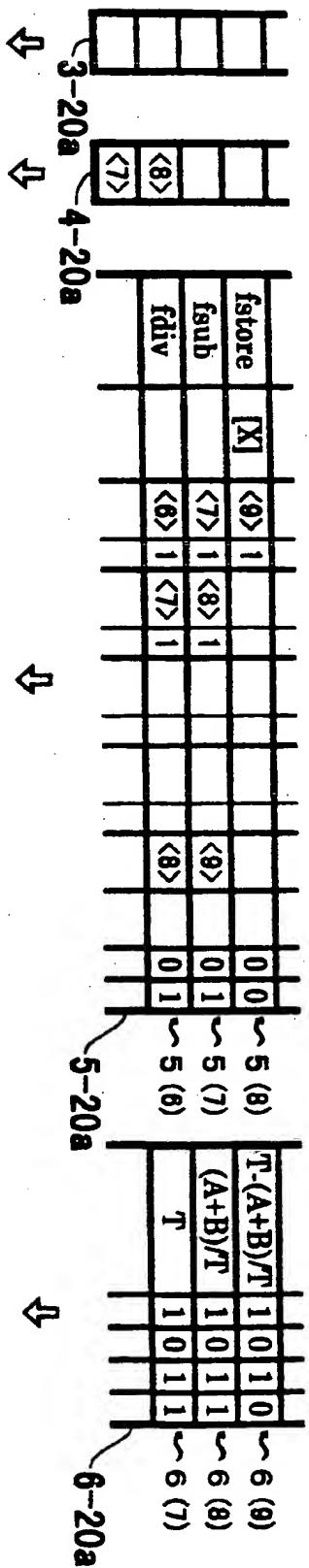
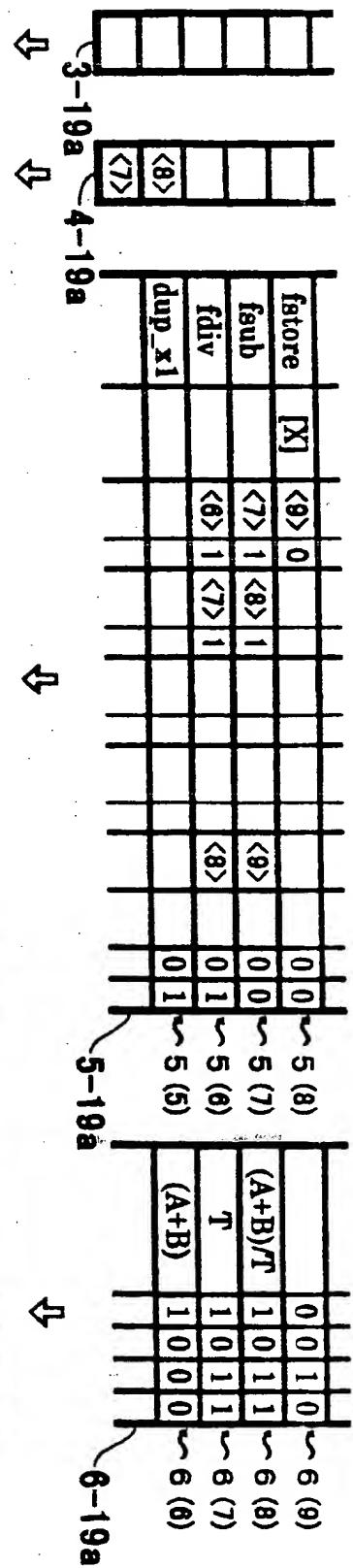
第 20 図

fstore	[X]	<9>	0	0 0	~ 5 (8)
fsub		<7>	1 <8>	1 0 0	~ 5 (7)
fdiv		<6>	1 <7>	1 0 1	~ 5 (6)
dup x1		<8>		0 1	~ 5 (5)
		<7>			

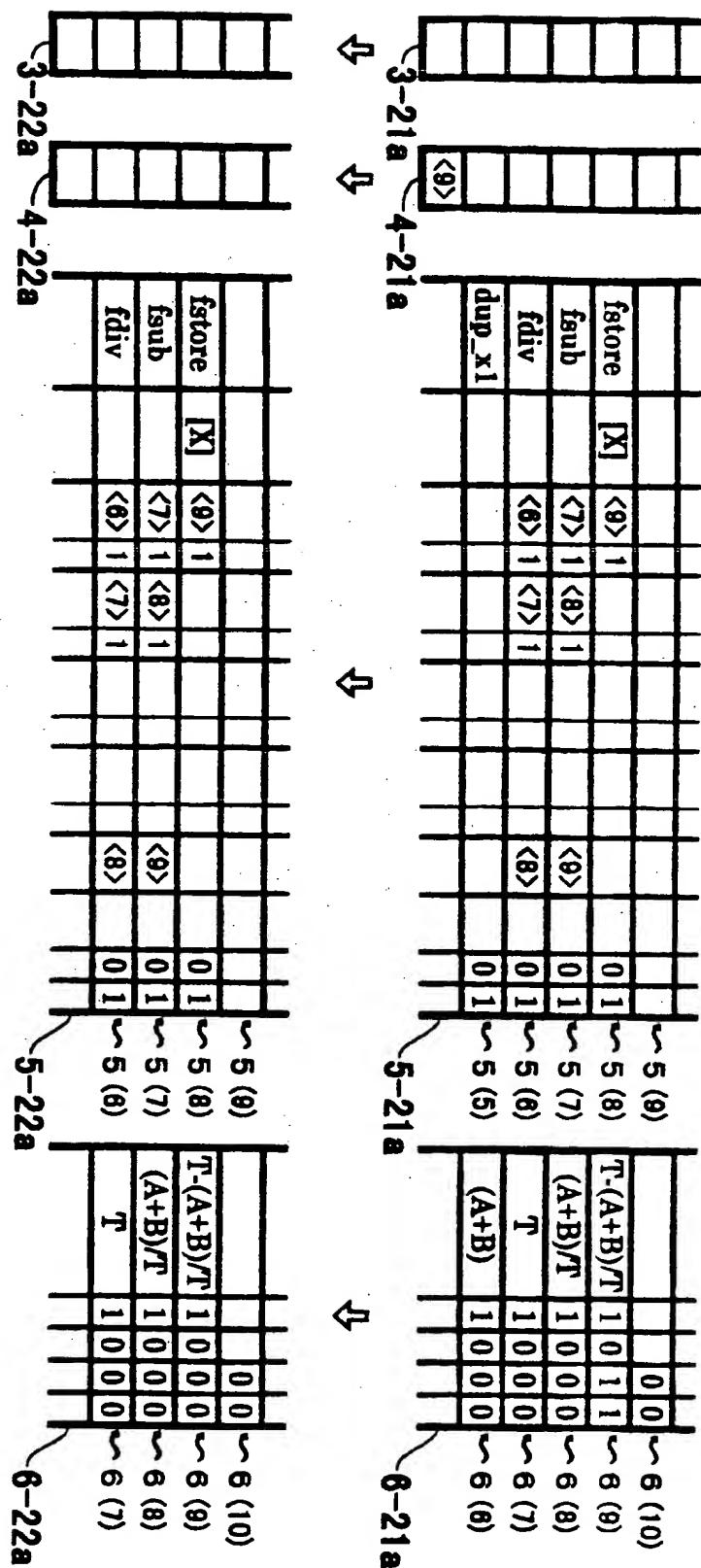


fstore	[X]	<8>	0	0 0	~ 5 (8)
fsub		<7>	1 <8>	1 0 0	~ 5 (7)
fdiv		<6>	1 <7>	1 0 1	~ 5 (6)
dup x1		<8>		0 1	~ 5 (5)
		<7>			

T-(A+B)/T		1 0 1 0	~ 6 (9)
(A+B)/T		1 0 1 1	~ 6 (8)
T		1 0 1 1	~ 6 (7)



第 21 図



A. CLASSIFICATION OF SUBJECT MATTER		Int'l. Cl. 6, G06F9/38, G06F15/82		Minimum documentation searched (classification system followed by classification symbols)		Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		Electronic base consulted during the international search (name of data base and, where practicable, search terms used)		C. DOCUMENTS CONSIDERED TO BE RELEVANT	
B. FIELDS SEARCHED		Int'l. Cl. 6, G06F9/38, G06F15/82		Minimum documentation searched (classification system followed by classification symbols)		Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		Electronic base consulted during the international search (name of data base and, where practicable, search terms used)		C. DOCUMENTS CONSIDERED TO BE RELEVANT	
Category*		Citation of document, with indication, where appropriate, of the relevant passages		Relevant to claim No.		A		JP, 2-260082, A (Hajime Seki), 22 October, 1990 (22. 10. 90) (Family: none)		1-6	
A		JP, 62-2330, A (NEC Corp.), 8 January, 1987 (08. 01. 87) (Family: none)		1-6		A		JP, 62-2330, A (NEC Corp.), 8 January, 1987 (08. 01. 87) (Family: none)		1-6	
Special categories of cited documents:		Later document published after the international filing date of priority date defining the general state of the art which is not considered to be of particular relevance		<input checked="" type="checkbox"/> See patent family annex.		Further documents are listed in the continuation of Box C.		<input type="checkbox"/> Further documents are listed in the continuation of the general state of the art which is not considered to be of particular relevance		Special categories of cited documents:	
* Special categories of cited documents:		Later document published after the international filing date of priority date defining the general state of the art which is not considered to be of particular relevance		<input checked="" type="checkbox"/> See patent family annex.		Further documents are listed in the continuation of Box C.		<input type="checkbox"/> Further documents are listed in the continuation of the general state of the art which is not considered to be of particular relevance		Special categories of cited documents:	
Date of mailing of the international search report		21 January, 1999 (21. 01. 99)		Name and mailing address of the ISA/JP/		Name and mailing address of the PCT/		Name and mailing address of the same patent family		Fax/phone No.	
Authorized officer		Japanese Patent Office		Telephone No.		Telephone No.		Fax/phone No.		Fax/phone No.	

第五版PC/T/ISA/210 (第二版一之二) (1998年7月)

This Page Blank (uspto)